

SANDIA REPORT

SAND97-1893 • UC-704

Unlimited Release

Printed July 1997

Development of Modifications to The Material Point Method for the Simulation of Thin Membranes, Compressible Fluids, and Their Interactions

Allen R. York II

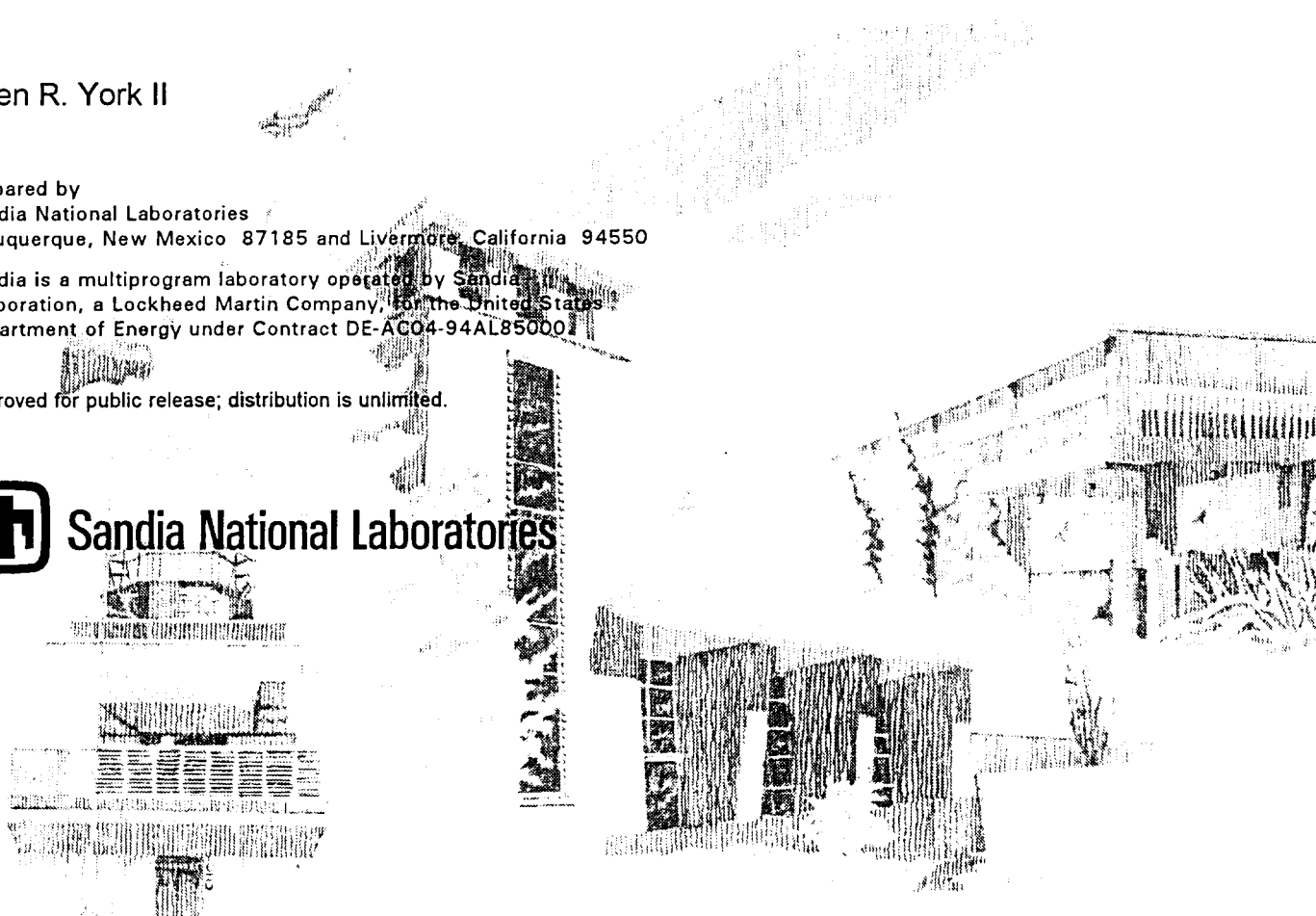
Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia
Corporation, a Lockheed Martin Company, for the United States
Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.



Sandia National Laboratories



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A08
Microfiche copy: A01

Development of Modifications to The Material Point Method for the Simulation of Thin Membranes, Compressible Fluids, and Their Interactions (pdf version)

Allen R. York II
Sandia National Laboratories
Engineering and Process Department
P.O. Box 5800
Albuquerque, New Mexico 87185-0483

Abstract

The material point method (MPM) is an evolution of the particle in cell method where Lagrangian particles or material points are used to discretize the volume of a material. The particles carry properties such as mass, velocity, stress, and strain and move through an Eulerian or spatial mesh. The momentum equation is solved on the Eulerian mesh. Modifications to the material point method are developed that allow the simulation of thin membranes, compressible fluids, and their dynamic interactions. A single layer of material points through the thickness is used to represent a membrane. The constitutive equation for the membrane is applied in the local coordinate system of each material point. Validation problems are presented and numerical convergence is demonstrated. Fluid simulation is achieved by implementing a constitutive equation for a compressible, viscous, Newtonian fluid and by solution of the energy equation. The fluid formulation is validated by simulating a traveling shock wave in a compressible fluid. Interactions of the fluid and membrane are handled naturally with the method. The fluid and membrane communicate through the Eulerian grid on which forces are calculated due to the fluid and membrane stress states. Validation problems include simulating a projectile impacting an inflated airbag.

In some impact simulations with the MPM, bodies may tend to “stick” together when separating. Several algorithms are proposed and tested that allow bodies to separate from each other after impact. In addition, several methods are investigated to determine the local coordinate system of a membrane material point without relying upon connectivity data.

Acknowledgments

This work was completed under the Ph.D. In-House Dissertation Program at Sandia National Laboratories, and their support is gratefully appreciated. I would like to express thanks to my supervisors Dr. Jerry Freedman and Robert Alvis for supporting me in this program. Also, thanks to Dr. Ned Hanson of my department at Sandia who encouraged me to stay in the program the times that I had doubts. My co-advisors Professor “Buck” Schreyer and Professor Deborah Sulsky deserve special thanks for the large amount of time they spent with me during this research. Also thanks to my committee members Doctors Attaway, Mello, and Brackbill and Professor Ingber.

TABLE OF CONTENTS

LIST OF FIGURES	VI
LIST OF TABLES	X
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. COMPUTATIONAL ASPECTS OF FLUID-STRUCTURE INTERACTION	5
2.1 Introduction to Eulerian, Lagrangian, and ALE Formulations	5
2.2 Fluid-Structure Simulation with Euler-Lagrange Coupling	9
2.3 ALE Methods for Fluid-Structure Interaction	13
2.4 Eulerian Methods for Fluid-Structure Interaction	14
2.5 Fluid-Membrane Interaction	15
2.6 Particle-In-Cell Methods	16
2.7 Immersed Boundary Methods	18
2.8 Stability of Fluid-Structure Interaction Formulations	19
2.9 Summary and Opportunities for Improvement in Current Approaches	20
CHAPTER 3. THE MATERIAL POINT METHOD	23
CHAPTER 4. THE MPM FOR FLUIDS AND VALIDATION OF THE FLUID FORMULATION	33
4.1 Simulation of Shock Propagation in a Fluid (Sod's Problem)	36
4.2 Gas Expansion	40
4.3 Chapter 4 Summary	41
CHAPTER 5. THE MPM FOR MEMBRANES AND VALIDATION OF THE MEMBRANE FORMULATION	43
5.1 Uniaxial Stress Formulation for a Spring or String	44
5.2 The One-Way Constitutive Equation	46
5.3 Computational Algorithm	46
5.4 Other Considerations	47
5.4.1 Resolving the Membrane Forces on a Cartesian Grid	47
5.4.2 Rotating Strains	48
5.4.3 Noise When Material Points Change Cells	48
5.4.4 Setting the Mass of the Membrane Points	49
5.4.5 Multiple Points through the Membrane Thickness	50
5.5 Spring-Mass System Simulation	50
5.5.1 Spring-Mass Problem Description	50
5.5.2 MPM Spring-Mass Vibration Simulation Results	51
5.6 String-Mass System With Initial Slack	54
5.7 Pendulum Simulation	56

5.7.1 Pendulum Problem Description	56
5.7.2 MPM Pendulum Simulation Results	57
5.7.3 MPM Pendulum Simulation Results - Without Wrinkle Algorithm.....	65
5.8 Plane Stress	65
5.9 Ball and Net Simulation	67
5.10 Chapter 5 Summary	70
CHAPTER 6. FLUID-STRUCTURE INTERACTION WITH THE MPM	71
6.1 Piston-Container Problem	72
6.1.1 Problem Description.....	72
6.1.2 Piston-Container Simulation Results	73
6.2 Membrane Expansion	75
6.2.1 Simulation 1	77
6.2.2 Simulation 2	77
6.2.3 Simulation 3	81
6.2.4 Simulation 4	81
6.3 Airbag Impact Simulation	88
6.3.1 Problem Description.....	88
6.3.2 Cyl-200 Simulation Results	89
6.3.3 Cyl-500 Simulation Results	91
6.4 Chapter 6 Summary	96
CHAPTER 7. CALCULATING NORMALS	97
7.1 Simple Color Function Approach	98
7.2 Quadratic Interpolation	98
7.3 Cubic Interpolation	99
7.4 Mass Matrix Approach.....	99
7.5 Testing the Methods on a Model Problem	100
7.5.1 Calculating Normals for the First Time Step	101
7.5.2 Simulating an Expanding Cylindrical Membrane.....	109
7.6 Further Analysis of the Mass Matrix Method.....	113
7.7 Using Quadratic Shape Functions in the Mass Matrix Method	123
7.8 Chapter 7 Discussion and Summary	127
CHAPTER 8. STABILITY ANALYSIS	129
8.1 Background	129
8.2 Introduction	129
8.3 Notation for Time Levels and Relevant Equations	130
8.4 Stability of the Material Point Equations	132
8.5 Numerical Simulations Performed to Test Analytical Results	141
8.6 Stability Summary.....	144
CHAPTER 9. SUMMARY AND CONCLUSIONS	145

REFERENCES	149
APPENDICES	153
A.1 CONTACT-RELEASE ALGORITHM	154
A.1.1 Background	154
A.1.2 Two-Bar Impact With No Contact-Release Algorithm	154
A.1.3 Algorithm to Allow Release	155
A.1.4 Calculating Grid Normals	158
A.1.5 Two-Bar Impact With Contact-Release Algorithm	160
A.1.6 Ball and Net Simulation	164
A.2 EXAMPLE: NOISE FROM A PARTICLE CHANGING CELLS	171
A.3 INTERPOLATION FUNCTIONS	173
A.3.1 Linear Interpolation	173
A.3.2 Quadratic Interpolation	174
A.4 INPUT FILES	176
A.4.1 Sod's Problem (Section 4.1, page 36)	177
A.4.2 Gas Expansion (Section 4.2, page 40)	178
A.4.3 Spring-Mass Simulation (Section 5.5, page 50)	179
A.4.4 String-Mass (Section 5.6, page 54)	180
A.4.5 Pendulum Simulation (Section 5.7, page 56)	181
A.4.6 Ball and Net Simulation (Section 5.9, page 67)	183
A.4.7 Membrane Expansion (Section 6.2, page 75)	184
A.4.8 Cyl-200 Airbag Simulation (Section 6.3, page 88)	185
A.4.9 Cyl-500 Airbag Simulation (Section 6.3.3, page 91)	186
A.4.10 Piston-Container (Section 6.1, page 72)	187
A.4.11 Stability Test Problem (Section 8.5, page 141)	189
A.4.12 Two-Bar Impact (Section A.1.5, page 160)	190
A.4.13 Ball and Net Simulation (Section A.1.6, page 164)	191
A.4.14 Maple Calculations	192

Table of Contents

List of Figures

Figure 1. Eulerian (a) and Lagrangian (b) Concepts	6
Figure 2. (a) An Eulerian and (b) A Lagrangian Impact Simulation	8
Figure 3. ALE and Lagrangian Calculation of a Bar (Huerta, 1994)	9
Figure 4. Euler-Lagrange Coupling Concept	10
Figure 5. PISCES 2DELK Axisymmetric Model for Fluid-Structure Interaction Simulation (Nieboer 1990)	12
Figure 6. Various Fluid-Membrane Interaction Simulations In The Literature	16
Figure 7. Mesh of Lagrangian Material Points and Their Subdomains	25
Figure 8. Mesh of Lagrangian Material Points Overlaid on the Computational Mesh	26
Figure 9. Material Point Convection	29
Figure 10. A Material Point in a Computational Cell	30
Figure 11. Sod's Fluid Shock Propagation Problem	36
Figure 12. Initial material point Positions for the MPM Simulation of Sod's Problem	37
Figure 13. Comparison of Density Calculations in Sod's Problem	38
Figure 14. Results of Sod's Problem Simulation with the MPM	39
Figure 15. Gas Expansion Problem	40
Figure 16. Gas Expansion Results from the MPM and FLIP-Material Point Positions	41
Figure 17. Gas Expansion Pressure Contours from the MPM and FLIP	42
Figure 18. Gas Expansion Specific Internal Energy from the MPM and FLIP	42
Figure 19. Three-Dimensional Representation of a Membrane (left) and Section View	43
Figure 20. (a) Points in MPM Simulation, (b) Physical Representation of Membrane and its Local Coordinate System and (c) Perspective View of Membrane Surface	44
Figure 21. One-Way Constitutive Form to Simulate Wrinkles in a String	46
Figure 22. Material Points Representing a Membrane Oriented Obliquely to the Grid	47
Figure 23. Solid: (a) Physical Representation and (b) MPM Representation and Membrane: (c) Physical Representation (d) MPM Representation	49
Figure 24. Idealized Spring-Mass System	51
Figure 25. Spring-Mass Vibration Simulation - Initial Positions	52
Figure 26. Time History of Mass Displacement (top) and Energy (bottom)	53
Figure 27. String-Mass with Initial Slack: (a) Physical and (b) MPM Representations	54
Figure 28. Particle Positions during the String-Mass Simulation	54
Figure 28. Material Point Positions at Various Times	54
Figure 29. Displacement and Energy Results for the String-Mass Simulation	55

Figure 30. Pendulum Problem Set-Up	56
Figure 31. Pendulum Problem Simulations - Initial Conditions for $\Delta =$ (a) 0.2, (b) 0.1, (c) 0.05, and (d) 0.025	58
Figure 32. Pendulum Position at Various Times - MPM Simulation (20 mp)	59
Figure 33. Pendulum Position at Various Times - MPM Simulation (40 mp)	60
Figure 34. Pendulum Position at Various Times - MPM Simulation (80 mp)	61
Figure 35. Pendulum Position at Various Times - MPM Simulation (320 mp)	62
Figure 36. Pendulum Position at Various Times - FEM Simulation	63
Figure 37. Comparison of Angle Theta For Swinging Pendulum Simulations	64
Figure 39. Plane Stress Assumptions	65
Figure 38. Pendulum Simulation (40 mp, $\Delta=0.1$) Without Wrinkle Algorithm	66
Figure 40. Ball and Net	68
Figure 41. Ball and Net Material Point Positions at Various Times	69
Figure 42. Center of Mass Velocities and Positions	69
Figure 43. Fluid-Structure Coupling	71
Figure 44. Piston-Container Problem	72
Figure 45. Piston-Container MPM Simulation Set-up	73
Figure 46. Mass (Piston) Deflection in the Piston-Container Simulations	74
Figure 47. Initial Conditions for the Dog-bone Membrane Expansion Simulation	75
Figure 48. Material Point Position Plots for Simulation 1	78
Figure 49. Radii and Energy for Membrane Expansion Simulation 1	79
Figure 50. Material Point Position Plots for Simulation 2	80
Figure 51. Radii and Energy for Membrane Expansion Simulation 2	82
Figure 52. Material Point Position Plots for Simulation 3	83
Figure 53. Radii and Energy for Membrane Expansion Simulation 3	84
Figure 54. Material Point Position Plots for Simulation 4	85
Figure 55. Radii and Energy for Membrane Expansion Simulation 4	86
Figure 56. Pressure and Membrane Stress Contours for Simulation 4	87
Figure 57. Airbag Impact Problem	88
Figure 58. Initial Configuration with the Internal Tether	89
Figure 59. Deformed Airbag Shapes of Cyl-200 Simulation	90
Figure 60. Displacement Results for Cyl-200 Simulation	91
Figure 61. Comparison of PISCES and MPM Deformed Configuration for $t=40$ ms	92
Figure 62. Deformed Airbag Shapes of Cyl-500 Simulation	93

Figure 63. Displacement Results for Cyl-500 Simulation	94
Figure 64. Comparison of PISCES and MPM Deformed Configuration for $t=40$ ms	95
Figure 65. Determination of Normal to Material Point p	97
Figure 66. (a) Quarter Symmetry Model Problem for Testing Normal Calculations and (b) Material Point Plots for No Fluid, 4 Points-Per-Cell (PPC) Fluid, and 16 PPC Fluid	101
Figure 67. Normals to the Membrane Points Calculated by the Simple Color Function Method for: (a) Vacuum and (b) 4 PPC Fluid Material Points	102
Figure 69. Normals to the Membrane Points Calculated by the Quadratic Method for 16 PPC Fluid Material Points	103
Figure 68. Normals to the Membrane Points Calculated by the Quadratic Method for: (a) Vacuum and (b) 4 PPC Fluid Material Points	104
Figure 70. Normals to the Membrane Points Calculated by the Cubic Method for: (a) Vacuum and (b) 4 PPC Fluid Material Points	105
Figure 71. Normals to the Membrane Points Calculated by the Cubic Method for 16 PPC Fluid Material Points	106
Figure 72. Normals to the Membrane Points Calculated by: (a) the Mass Matrix Method and (b) Using the Material Point Connectivity	108
Figure 73. Change in Cylinder Radius	109
Figure 74. Material Point Positions and Normals for: (a) the Cubic Method and (b) the Quadratic Method at $t=0.01$	111
Figure 75. Material Point Positions When Calculating Normals Using (a) Connectivity, (b) Mass Matrix (linear shape functions), (c) Cubic, and (d) Quadratic Methods	112
Figure 76. Membrane Points Oriented at 45 deg, Associated Color Contours, and Calculated Normals	114
Figure 77. Membrane Points Oriented at 10 deg, Associated Color Contours, and Exact Calculated Normals	115
Figure 78. Membrane Points Along a Curve $xy=C$, Associated Color Contours, and Calculated Normals	116
Figure 79. Membrane Points Along a Circular Arc and Associated Color Contours, and Exact and Calculated Normals	117
Figure 80. Cases Where the Method to Breaks Down: (a) Nearly Singular Mass Matrix, and (b) Nearly Symmetric Grid Color Solution that Results in No Gradient at the Cell Center	118
Figure 81. Nomenclature	119
Figure 82. Unit Cell with Three Material Points	120
Figure 83. Normals to a Circular Arc	121
Figure 84. Normals to a Circular Arc	122
Figure 85. Normals to a Circular Arc Using Quadratic Shape Functions	124
Figure 86. Normal Vectors Calculated Using: (a) Mass Matrix and (b) Connectivity	125

Figure 87. Membrane Expansion Simulation Using Connectivity and the Mass Matrix Method to Determine Normals	126
Figure 88. Ghost Cell Concept	127
Figure 89. One-Dimensional Problem Notation	132
Figure 90. Shape Functions Associated with Each Element	133
Figure 91. Numerical Results of Stability Tests for: (a) Points Initially in the Cell Centers and (b) Points Initially at the Sides of the Cells	143
Figure 92. Initial Conditions for Two-Bar Impact	154
Figure 93. Two-Bar Impact Simulation with the MPM: (a) Material Point Positions, and (b) Center of Mass Velocities and Positions	156
Figure 94. Grid Normals for One Bar of the Two-Bar Impact Simulation	160
Figure 95. Two-Bar Impact Simulation with the MPM and the Contact-Release Algorithm: (a) Material Point Positions, and (b) Center of Mass Velocities and Positions	161
Figure 96. (a) Energy History for Two-Bar Impact and (b) Comparison of Energy History for Simulations with $v_1^0=20$	163
Figure 97. Ball and Net	164
Figure 98. Ball and Net Simulation Results Without Contact-Release Algorithm	165
Figure 99. Ball and Net Material Point Plots With Contact-Release Algorithm	167
Figure 100. Ball and Net Velocity/Position and Energy Results	168
Figure 101. Ball and Net Normals at Time $t \sim 1$	169
Figure 102. Set Up for a MPM Spring Simulation	171
Figure 103. Forces and Stresses Prior to Material Point Crossing a Cell Boundary	171
Figure 104. Forces and Stresses After Material Point 6 Crosses a Cell Boundary	172
Figure 105. Nomenclature	173
Figure 106. (a) Nomenclature and (b) Neighbor Cell Designations	174

List of Tables

Table 1: Pendulum Simulation Parameters	56
Table 2: Parameters for the Ball and Net Simulation	68
Table 3: Piston-Container Periods of Vibration	74
Table 4: Fluid and Membrane Properties	75
Table 5: Membrane Expansion Simulation Parameters and Results	76
Table 6: Comparison of Hoop Stress	77
Table 7: Airbag Impact Simulations	88
Table 8: Airbag Impact Simulation - Material Properties	88
Table 9: Material Point Method - Equations for One Full Time Step	131
Table 10: Stability Test Problem Parameters	141
Table 11: Parameters for the Ball and Net Simulation	164
Table 12: Quadratic Interpolation Functions for a Unit Cell in Logical Space	175
Table 13: Derivatives Quadratic Interpolation Functions	175

CHAPTER 1. INTRODUCTION

There is an important class of problems involving fluid-structure interaction, as exemplified by airbag deployment in automobiles, for which a robust numerical algorithm would be desirable. The existing approaches are limited mainly due to the general problem of treating interfaces. Some of the interface problems include determining the boundaries of the fluid and structure and applying the correct conditions on one from the other. The material point method (MPM) developed for other classes of problems appears to hold considerable promise for this class as well. The objective of this research was to modify the MPM for fluid-membrane interaction problems and to investigate its potential usefulness.

In the MPM, Lagrangian particles or points are used to discretize the volume of the fluid or solid. These material points carry with them properties such as mass, velocity, stress, and elastic and plastic strain. The points move through an Eulerian mesh on which governing equations are solved and derivatives are defined.

The MPM has several advantages. Mesh distortion or entanglement is not a problem since the Eulerian mesh is under user control. Thus, highly distorted configurations of material points can be simulated whether it is gas flow or deformation of a solid. Since all properties are carried by the material points, numerical dissipation through the fixed mesh is small. Interfaces and material boundaries are defined by the location of the Lagrangian material points so that boundary reconstruction and mixed-cell calculations, that may be computationally intensive in a pure Eulerian method, are not needed. The MPM has inherent in its formulation an automatic calculation of no-slip contact. Thus, for a large class of problems no additional contact calculations are needed. Slip and friction can be simulated with the addition of a contact algorithm.

The method is potentially attractive when considering fluid-structure interaction. The motion of the fluid material points and solid material points are governed by the same equations. The difference between fluid and solid points is only in the evaluation of the respective constitutive equations and the additional solution of the energy equation for fluid points.

Chapter 2 reviews the three basic computational approaches consisting of the Eulerian, Lagrangian, and Arbitrary Lagrangian-Eulerian (ALE) methods and provides a summary of current literature. It is evident that one of the areas where the proposed method improves upon others is in

handling interfaces.

Chapter 3 summarizes and develops the material point method in the finite element framework. The conservation equations for momentum, energy, and mass are discussed. Here, the framework is laid for subsequent modifications to the method.

In Chapter 4 modifications to the MPM are proposed that enable fluids to be simulated. The numerical algorithms for the fluid momentum, constitutive, and energy equations are developed and validation problems are provided. Sod's problem of a shock propagating through a compressible gas is solved, and the results compare well with theory. A gas expansion problem is solved, and the results are compared with those from another fluid dynamics code.

In Chapter 5, development of the membrane material point that allows simulation of springs, strings, and membranes is discussed. A one-way constitutive equation is proposed to allow wrinkles in a membrane; that is, a material point may accumulate compressive strain without compressive stress. Thus, an approximation to a small wrinkle is achieved without resolving the physical geometry of the wrinkle which would require a fine computational grid. Simulations of a spring, string, and swinging pendulum demonstrate the method. Sequential mesh refinement for the pendulum problem demonstrates convergence.

The proposed method for fluid-structure simulations is presented in Chapter 6. An arbitrarily-shaped membrane filled with a viscous gas under pressure that is not initially in equilibrium is simulated. Large oscillations in the membrane are eventually damped out, and the membrane assumes a circular shape. The gas does not escape from the membrane, and the membrane (hoop) stresses are consistent with the final pressure of the gas. Convergence is demonstrated in four consecutive simulations. A simulation of a pre-inflated airbag being penetrated by a probe is also simulated. The deflection of the probe into the bag agrees well with other simulations and experiment.

Chapter 7 presents various methods for determining the normal vectors to membrane points. The normal vectors must be determined to evaluate the membrane constitutive equation. The connectivity of material points can be used to find normal vectors, but a more general method is preferred. One method is shown to be superior and is successful when used on a model problem.

A stability analysis for linear elastic material points is presented in Chapter 8. A general stability analysis of the method is difficult because of the Lagrangian step and remap involved in a single com-

putational cycle. The complexity of the equations is even greater if material points are allowed to cross cell boundaries. Thus, a simplified one-dimensional analysis is presented that indicates the Courant-Friedrichs-Lewy (CFL) stability condition governs where the length scale is based on the mesh and not the material point spacing (Courant et al., 1928). This is no surprise as numerical results have consistently shown this to be the case. It should be noted that a stability analysis has not been conducted for the fluid formulation or the fluid-structure interaction formulation. However, numerical simulations have indeed shown that the CFL condition also governs the time step.

Section [A.1](#) describes a solution to one of the problems discovered during the course of the research. In some simulations where bodies impact there is a nonphysical sticking of the two bodies when they should be moving away from one another. A description is given of the contact-release algorithm whereby grid velocities from different bodies are compared to determine if they should be in contact or not. The algorithm is successfully applied to two impacting bars and to a ball impacting a net.

The result of this research is a simple explicit algorithm based on the MPM for simulating membranes, compressible fluids, and fluid-structure interaction. The potential usefulness of the algorithm is demonstrated by solving a wide variety of problems. In most cases the results of the demonstration problems are compared to theory, experiment, or to another simulation, and the results from the MPM consistently agree well with analytical solutions and numerical solutions based on alternative approaches.

CHAPTER 2. COMPUTATIONAL ASPECTS OF FLUID-STRUCTURE INTERACTION

Fluid-structure interaction is a broad class of problems where the combined states of a fluid and a solid need to be determined simultaneously. These problems are especially important when the structural response is non-trivial (e.g., not rigid) and, in turn, has an effect on the fluid. Some examples are airflow around high-speed reentry vehicles, airflow around parachutes, fluid sloshing in a storage tank, and airflow in an inflating airbag.

When computational power was limited, the fluid parts of the fluid-structure problem were often solved with major simplifying assumptions for the structural part of the problem. Similarly, if the structural aspect of the problem was considered to be dominant, simplifications were made for the fluid behavior. However, today computers and methods have advanced to the point where the combined problem may be solved, although there is still much room for improvement.

If the structure part of the fluid-structure problem is a solid, the constitutive equation depends on deformation which is usually small. The fluid, which may undergo large deformations, has a constitutive equation that involves deformation rate and/or a constituent model such as incompressibility. The boundary between the fluid and structure is where the complex interaction phenomena occur.

For an inviscid fluid the boundary condition on the fluid at the fluid-structure interface should be one of free slip. For viscous fluids the condition may be no slip at the interface. However, the normal component of traction in the fluid should always be the same as that in the structure.

For numerical solutions of fluid-structure problems, an additional complication occurs when the solid has one small dimension, which leads to a time step restriction if the structure is treated as a continuum or leads to the need for plate or shell theories and corresponding difference or finite element schemes. For a fluid-membrane problem there are further problems. Since the structure has no bending stiffness the deformations can be large even though the strains may be small.

2.1 Introduction to Eulerian, Lagrangian, and ALE Formulations

The Eulerian method uses a mesh fixed in space where the nodes represent the discretized spatial variable. On the other hand, Lagrangian methods use the location of material points at a previous step as the spatial variable. Total Lagrangian methods use positions at a reference time, and numerical schemes based on this method also use a fixed mesh. However, updated Lagrangian methods are more commonly used where the location of a material point at the end of the last time step is repre-

sented by nodes of a mesh.

In Eulerian formulations the governing equations are solved on the fixed grid, and material moves through the mesh. In Lagrangian methods, the computational grid or mesh is attached to the material being simulated. The Lagrangian mesh moves and distorts with the material.

Figure 1 illustrates these concepts with a simple example of a material loaded with pressure and constrained on the right and bottom sides. In the Eulerian method (Fig. 1(a)) material is represented by quantities defined at the grid nodes, such as mass. At some later time, t_1 , in the simulation, the mass at the grid nodes has changed (convected) due to material deformation as governed by external and internal forces, but the grid remains fixed in space. The changes in the sizes of the circles indicate the change in the amount of mass at the grid nodes. An attempt to describe the mass distribution is made by showing grid masses associated with material boundaries designated with dashed lines. In Fig. 1(b), which shows the Lagrangian approach, the material is represented by the grid which distorts with the material.

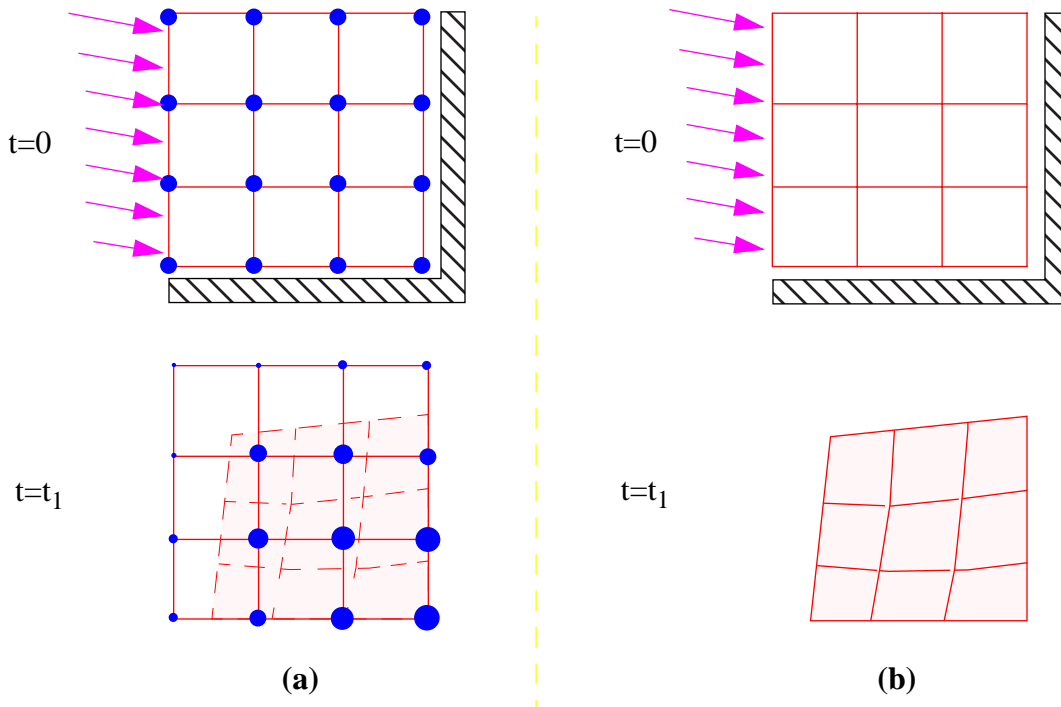


Figure 1. Eulerian (a) and Lagrangian (b) Concepts

Figure 1 shows one disadvantage of the Eulerian method, which is the lack of definition of mate-

rial boundaries. If a material boundary exists in one cell, then the mass at one grid node may be large, while another may be small, and the actual material boundary is somewhere between the two nodes.

Another disadvantage of Eulerian methods has been the lack of ability to resolve material interfaces when two or more materials are in the same computational cell. Also, because the grid is fixed, history-dependent or path-dependent materials (e.g., viscoelastic fluids or plastic solids) have been difficult to simulate.

In an Eulerian formulation, because the grid is fixed, a nonlinear convective acceleration term must be included in the governing equations. This nonlinear term is generally expensive to handle from a computational perspective. To overcome this limitation some approaches split the Eulerian calculation into separate Lagrangian and Eulerian (remap) steps.

The great advantage of Eulerian methods is the ability to simulate highly distorted fluid flow and large deformations. Eulerian methods are not limited by mesh deformation. Also, free surfaces can be created in an Eulerian method “automatically” whereas this is more difficult in a Lagrangian code.

Lagrangian methods, however, do not have a nonlinear convective term as a part of the governing equations. Thus, the computations are less complicated in general until the mesh gets so distorted that remeshing is required. Lagrangian methods do not require special procedures to resolve material interfaces, and history-dependent materials are simple to model.

Usually special features have to be implemented into both Lagrangian and Eulerian codes for simulating impact between materials.

A recent application of an Eulerian method was the simulation of impact of the Shoemaker-Levy Comet on Jupiter with the code CTH (Deitz, 1995; Hertel, 1993). This code is used to simulate high velocity impacts as illustrated in Fig. 2(a) where a copper ball penetrates a steel plate at a velocity of 14,760 ft/s (4,500 m/s). Note the numerous regions involving material failure with resulting free surfaces.

An example of a Lagrangian calculation done with PRONTO3D (Taylor, 1989) is shown in Fig. 2(b) where a shipping container impacts a rigid target at 200 ft/s (76 m/s) (Slavin, 1994). Note the distorted elements near the impact point at the bottom of the container. In this simulation, several contact surfaces are defined to simulate sliding and contact between materials (Heinstein et al., 1993).

The arbitrary Lagrangian-Eulerian (ALE) method is formulated so that the mesh can either move

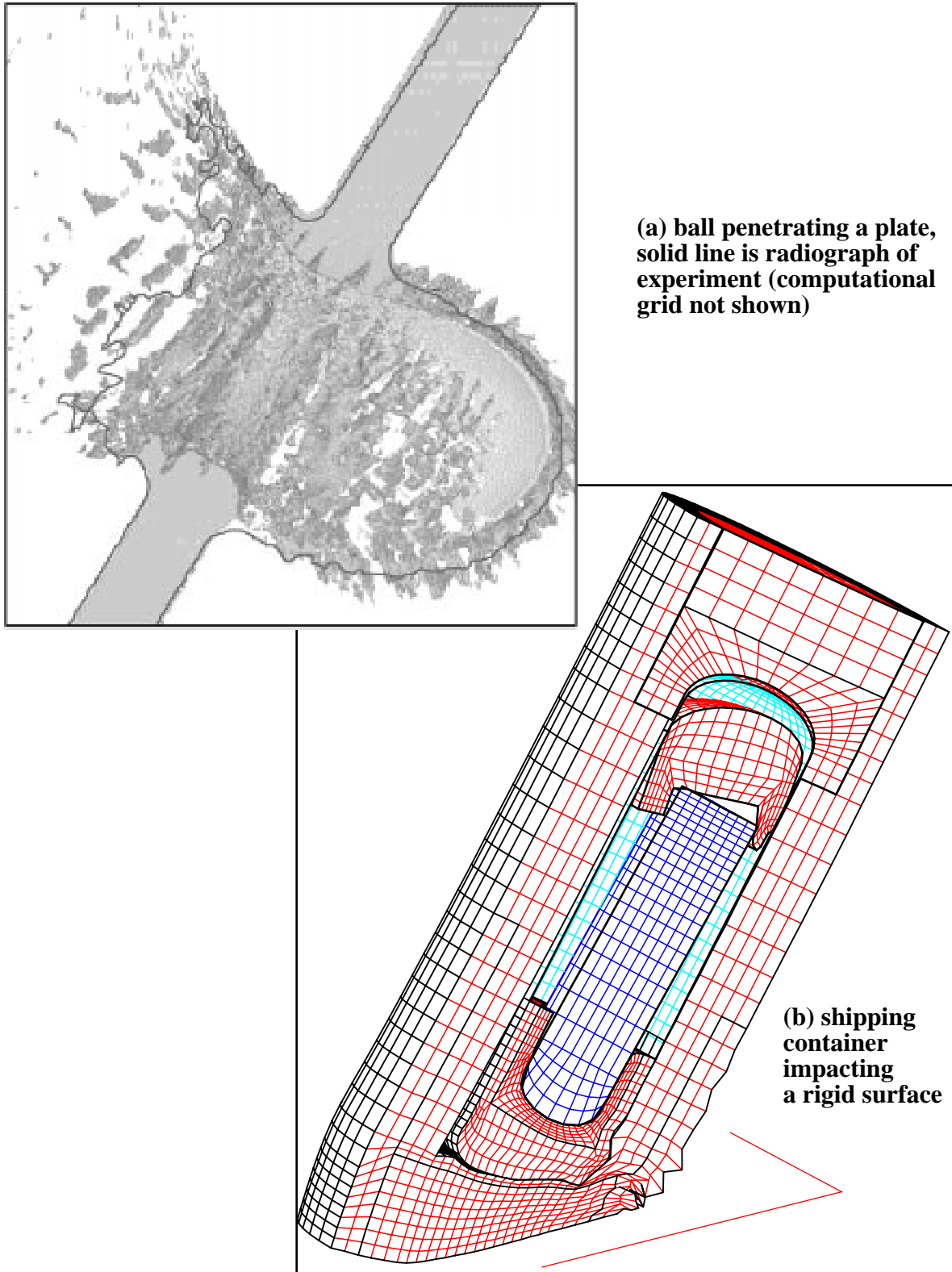


Figure 2. (a) An Eulerian and (b) A Lagrangian Impact Simulation

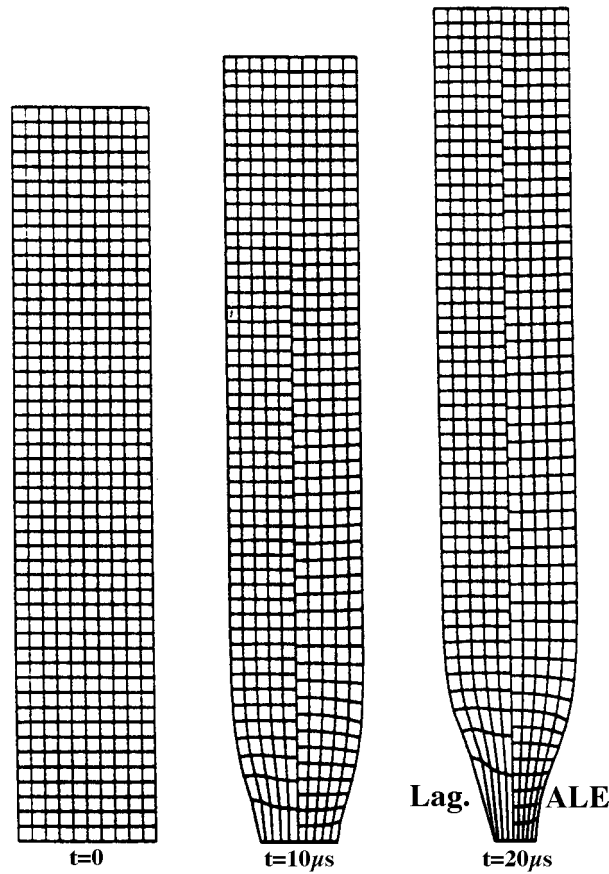


Figure 3. ALE and Lagrangian Calculation of a Bar (Huerta, 1994)

with the material, remain fixed in space, or move at an arbitrary velocity. The mesh can be kept regular during the calculation if the proper mesh velocity is specified. Also, the mesh may follow or be adapted to certain discontinuities (e.g., a shock) to improve accuracy or resolution. Of course, one of the “tricks” of the method is to specify the “right” mesh velocity. Figure 3 illustrates an ALE versus a Lagrangian calculation of the necking that occurs in a bar being pulled. Notice that the mesh is more regular and, presumably, the results are more accurate in the highly deformed region for the ALE calculation (Huerta, 1994).

2.2 Fluid-Structure Simulation with Euler-Lagrange Coupling

One of the most obvious ways to solve a problem that involves both fluid and solid materials is to couple two existing codes - one for fluid-only simulation and one for solid-only simulation. This method may take advantage of advanced features that exist in each of the separate codes.

McMaster (1984) describes coupling between fluid and solid dynamics codes for simulation of

fluid-structure interaction. In one application the coupling is such that the structure applies a position and velocity boundary condition to the fluid, and the fluid applies a pressure boundary condition to the structure. This is typical of most coupling methods. The algorithm iterates until the boundary conditions and incompressibility condition are satisfied. That is, the fluid pressure field and position of the structure are corrected until the normal velocities of the fluid and structure are equal. Prior to the iteration, the intersection of the structure with the Eulerian cells must be defined.

Figure 4 illustrates the coupling concept. On the left, velocity compatibility is symbolically represented showing the velocity vectors of the Lagrangian shell and the Euler grid nodes. The normal velocities of the fluid and solid must be equal to prevent penetration of the fluid through the solid. Thus, the velocity of the shell nodes must be mapped to the Euler grid nodes in a way that accurately represents the no penetration condition. The force compatibility part of Fig. 4 shows the forces applied on the shell by the fluid. A value of pressure at appropriate Euler grid nodes is used along with the information of where the shell intersects the Euler grid to determine a magnitude and direction of force that should be applied to the shell. Pressures that may be used in this calculation are listed P_1 - P_6 .

As a model problem to illustrate the coupling, McMaster provided the simulation of a cylinder

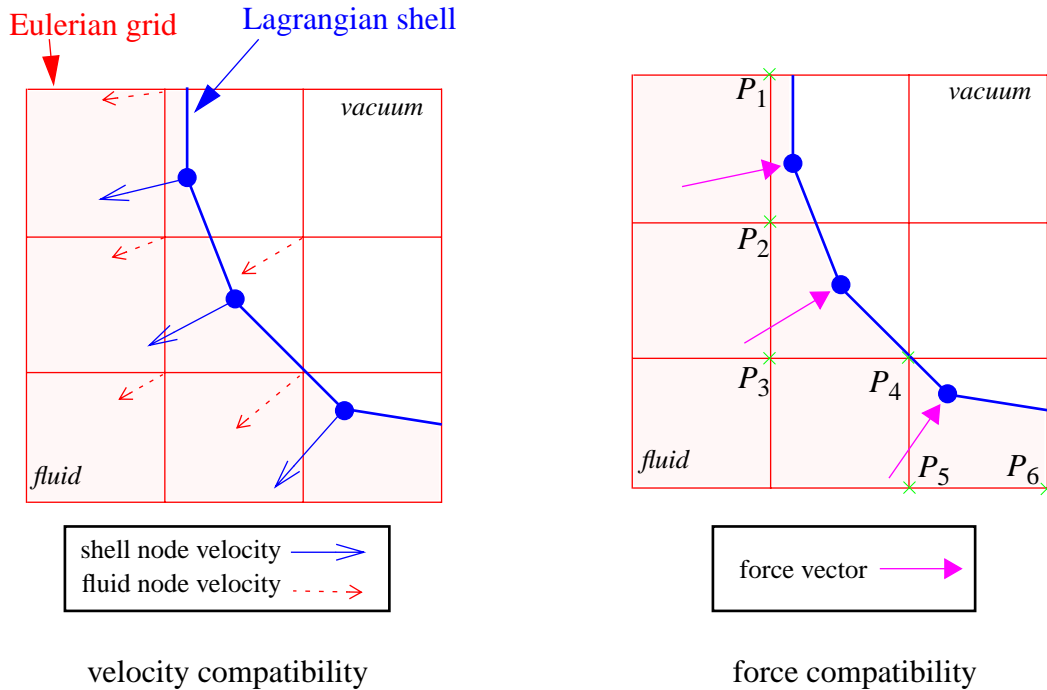


Figure 4. Euler-Lagrange Coupling Concept

subjected to mode 2 vibration while submerged in water.

In a different application, McMaster describes the coupling of a compressible explicit Eulerian fluids code with DYNA. The coupling method is similar to that described above. It is noted that special provisions are implemented to handle fluid advection when the structure is present in a cell. Also, massless marker particles can be used to monitor the flow field. A simulation of a sphere impacting water shows good agreement with experimental data.

A similar coupling is described by Gross (1977) between two explicit codes. Several steps are taken in the structural code for each step in the fluid code. This is called sub-cycling. Cylinders impacting water are simulated.

MSC/DYTRAN is an explicit finite element program (Buijk, 1991, 1993; Florie, 1991) that has been used to simulate gas and material dynamics of unfolding automobile airbags. The gas dynamics portion of the simulation can be turned on or off. However, when the gas flow is simulated, the intersection of a membrane (airbag material) element and the Euler element is calculated, and the pressure within the Euler element is distributed among the nodes of the membrane elements as forces. Buijk terms this treatment as a general Euler-Lagrange coupling because of the unlimited relative motion of the Lagrangian membrane elements to the Eulerian elements. This is in contrast to ALE (Arbitrary Lagrange-Euler) coupling where the two elements are connected and the motion of the Lagrangian elements is limited. Buijk reports that most of the cpu time for a simulation including gas dynamics is used to determine the intersection of the Lagrangian membrane elements and the Eulerian elements. This is essentially a contact algorithm to determine contact between the gas and the membrane.

A simulation of the impact of an unfolding airbag with a plate shows favorable agreement with experiment. By running simulations with and without the gas dynamics, Buijk concluded that the momentum of the gas plays a significant role prior to full inflation of the bag. That is, an out-of-position occupant who contacts the inflating airbag will have significant additional acceleration due to the gas momentum.

The PISCES 2DELK (**E**uler-**L**agrange **K**oupling) code provides coupling of a Lagrangian shell discretization with an Euler discretization for the fluid (gas) flow (Nieboer, 1990; de Co0, 1989; Prasad, 1989). This code was the predecessor of the MSC/DYTRAN code mentioned above, and thus, uses similar methods.

Nieboer (1990) describes PISCES simulations where an axisymmetric model was used to show the effect of a rigid body impacting a sealed airbag. The model is similar to that shown in Fig. 5. After the impactor contacts the airbag, the gas flow within the airbag is calculated using the Euler grid. The airbag is modeled using linear elastic membrane elements. Good agreement is observed between the simulation results and the experiments.

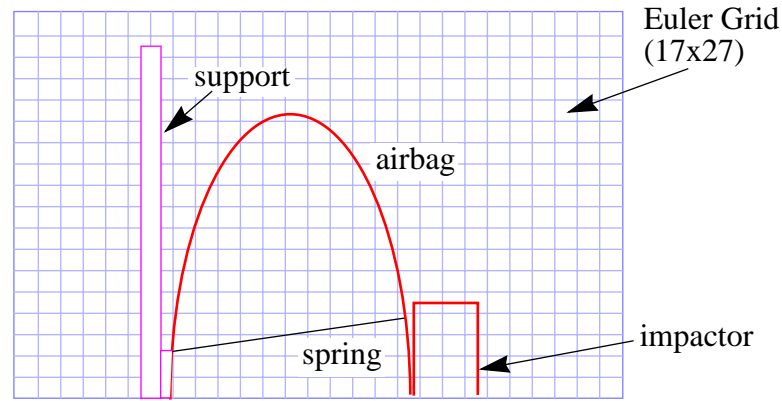


Figure 5. PISCES 2DELK Axisymmetric Model for Fluid-Structure Interaction Simulation (Nieboer 1990)

Lewis (1994) describes the coupling between NIKE3D, a solid dynamics code, with an ALE fluid dynamics code. The motivation for this approach is to exploit the advanced features (e.g., constitutive models) that may reside in each separate code. The scheme is iterative and implicit, and the time step in the fluid and solid domain is the same. A model problem was solved where an elastic cylinder is submerged in an inviscid, incompressible fluid. The fluid extends to a prescribed radial boundary. An axisymmetric pressure pulse on the inside of the cylinder sets it into the fundamental breathing mode of vibration. The predicted vibrational frequency compares well with the theoretical frequency. Other problems involving an underwater explosion and bubble dynamics show reasonable agreement with experimental data.

Ghattas (1994) describes a fluid-structure interaction method that uses an Eulerian fluid description and a Lagrangian structure description. The method was formulated specifically for fluid-structure interaction simulations and is not a coupling of two “separate” programs. A variational formulation is used, and a continuous traction and velocity field is required at the fluid-structure interface. The resulting set of nonlinear equations are solved with a Newton-like method. Ghattas illus-

trates the method by calculating two-dimensional deformations in elastic bodies due to a surrounding flow field.

Bendiksen (1994) indicates aerodynamic stability calculations can be affected by the errors accumulated when fluid-structure interaction is simulated by a “separate” sequential fluid and structure calculations (integrations). Bendiksen treats the equations for the fluid and structure as one problem by formulating the governing equations for both the fluid and structure in integral conservation-law form based on the same Eulerian-Lagrangian description. Several flutter problems are solved.

2.3 ALE Methods for Fluid-Structure Interaction

One of the first papers to describe the ALE method was by Hirt (1974). Hirt’s formulation involved a finite difference framework with an implicit time integration algorithm. Applications were for fluid dynamics simulations at all flow speeds. The computations to advance the solution one time step are separated into three phases. The first is an explicit Lagrangian calculation, with the exception that the mesh vertices do not move. The second is an iterative phase that adjusts the pressure gradient forces to the advanced time level. This optional phase eliminates the Courant stability condition. The mesh vertices are moved to their new Lagrangian positions after this second phase. The third phase, also optional, moves the mesh to a new position. In the Lagrangian phase, if the mesh has some velocity other than the one obtained by moving the mesh to its new position, convective fluxes must be calculated. This is often called rezoning. Hirt notes that the separation of the calculations into Lagrangian and convection phases originated in the Particle-in-Cell method. This is also called an Operator Split method and is described by Benson (1992, p. 325).

Donea (1977) describes the use of the finite element method for solving coupled hydrodynamics-structures problems. Donea’s method is conceptually similar to ALE methods that were originally based on finite differences. Here, the finite elements may move with the material, remain fixed, or move at an arbitrary velocity. The motivation for this approach is that it allows for a simple computer program architecture, permits straightforward treatment of fluid-solid interfaces, and enables the use of arbitrarily shaped elements for modeling both the fluid and solid. One fluid-structure problem is solved with both the Lagrangian and Eulerian methods. In the Eulerian method, the elements that are adjacent to the structure are prescribed to stay in contact with the structural elements, so that these elements are actually Lagrangian. The results for the two methods are nearly identical.

A more recent paper by Donea (1983) illustrates the use of the ALE finite element method in simulating a contained explosive detonation in water. Similar to that described above, fluid elements adjacent to structural elements are pure Lagrangian and stay attached to the structural elements. Donea gives a detailed description of how the accelerations are determined for nodes shared by fluid and structural elements that result in common normal velocities.

There are many other examples of ALE applications in the literature. Liu and Ma (1981) and Huerta (1990) use an ALE finite element method to simulate fluid sloshing in a tank. Nomura (1994) also uses an ALE finite element method to investigate flow-induced vibrations of a cylinder.

2.4 Eulerian Methods for Fluid-Structure Interaction

Eulerian methods have been and still are popular for fluid dynamics simulations due to their ability to handle large distortions. There are many papers in the literature on Eulerian methods for fluid dynamics, and many theoretical and practical problems have been solved with this method.

There are some problems traditionally associated with Eulerian methods. These problems are: (i) handling multiple materials in a cell, (ii) handling history-dependent materials, (iii) tracking material interfaces, and (iv) handling impact or contact between materials. However, these problems are being overcome with sophisticated advection and interface tracking algorithms. In Benson's (1992) survey, he states that most of the interface tracking algorithms use marker particles at surfaces or derive the surface definitions from the volume fractions of the different materials.

A state-of-the-art code called CTH is a two-step, second-order accurate Eulerian solution algorithm used to solve multi-material problems involving large deformations and/or strong shocks (Hertel, 1993). Material strength is included in the solution. Material contact is not resolved or simulated in a way that is possible with an ALE or Lagrangian code. The information available in a computational cell includes the amount of each material in the cell, so it is difficult to resolve boundaries and enforce contact conditions. History-dependent materials (e.g., plasticity) can be modeled. History variables are advected with a conservative, second-order accurate van Leer (1977) scheme.

No recent papers are found in the literature on using pure Eulerian methods for fluid-structure interaction where the interface of the fluid and structure needs to be resolved as a part of the solution.

2.5 Fluid-Membrane Interaction

A subset of fluid-structure problems is that of fluid-membrane interaction. Some practical examples involving fluid-membrane interaction are: (i) inflation of airbags, (ii) sailboat sails, (iii) flexible fabric roofs for buildings, (iv) moving automobile belts, (v) textiles (fiber and fabric production), (vi) paper manufacturing, (vii) VCR and other moving tape media, (viii) bands (such as bandsaws and automatic transmissions), and (ix) pressure transducers.

Niemi et al. (1987) use the finite element method to strongly couple gas and membrane dynamics to study natural frequencies of a membrane moving in air. In this case, the physical situation being modeled is that of paper moving between two rolls (Fig. 6a). Standard finite element techniques are used to obtain discrete equations of motion for the fluid and the membrane. The coupling is achieved by enforcing equality of normal (to the membrane) accelerations for “wet” nodes on the membrane. The resulting set of equations is used to calculate frequencies of vibration.

A variety of large-motion, fluid-membrane problems are solved by Han et al. (1987) using an iterative method. The fluid is assumed to be inviscid and irrotational, and is simulated with a boundary element method. The membrane is modeled using shell finite elements. The membrane and fluid elements are implemented in three-dimensional space. An analysis begins with a given membrane shape as a boundary to the fluid. The flow analysis is performed with the boundary element method. With the new fluid loads known, the finite element solution determines the position of the membrane. If the new membrane position is within a specified tolerance of the old profile the calculation continues to the next time step. If not, then the boundary element solution is called again given the new membrane position. Iterations continue until convergence is achieved. Good results are obtained for a pre-tensioned, pressure-loaded square membrane and a pressurized cylindrical membrane in crossflow (Fig. 6b).

Yamamoto et al. (1992) simulate steady two-dimensional flow past a flexible membrane using finite difference techniques (Fig. 6c). The fluid is governed by Navier-Stokes equations in terms of a vorticity-stream function. Coupling of the fluid with the membrane is done by enforcing zero normal fluid velocity at the surface of the membrane and by applying forces to the membrane due to the fluid pressure.

Smith (1995) has simulated incompressible, unsteady viscous flow over a flexible, linear elastic membrane wing with an implicit finite difference method (Fig. 6d). The fluid imparts a normal and

shear stress to the membrane. The boundary conditions at the membrane surface require the fluid velocity to equal the membrane velocity. An iterative procedure is used to solve the coupled problem until a predetermined convergence criterion is satisfied.

Other papers in the literature use methods similar to those described above to simulate cylindrical pneumatic structures subjected to wind loading (Uemura, 1971), blood flow (Rast, 1994; Fig. 6e), and membrane sensors (Lerch, 1991).

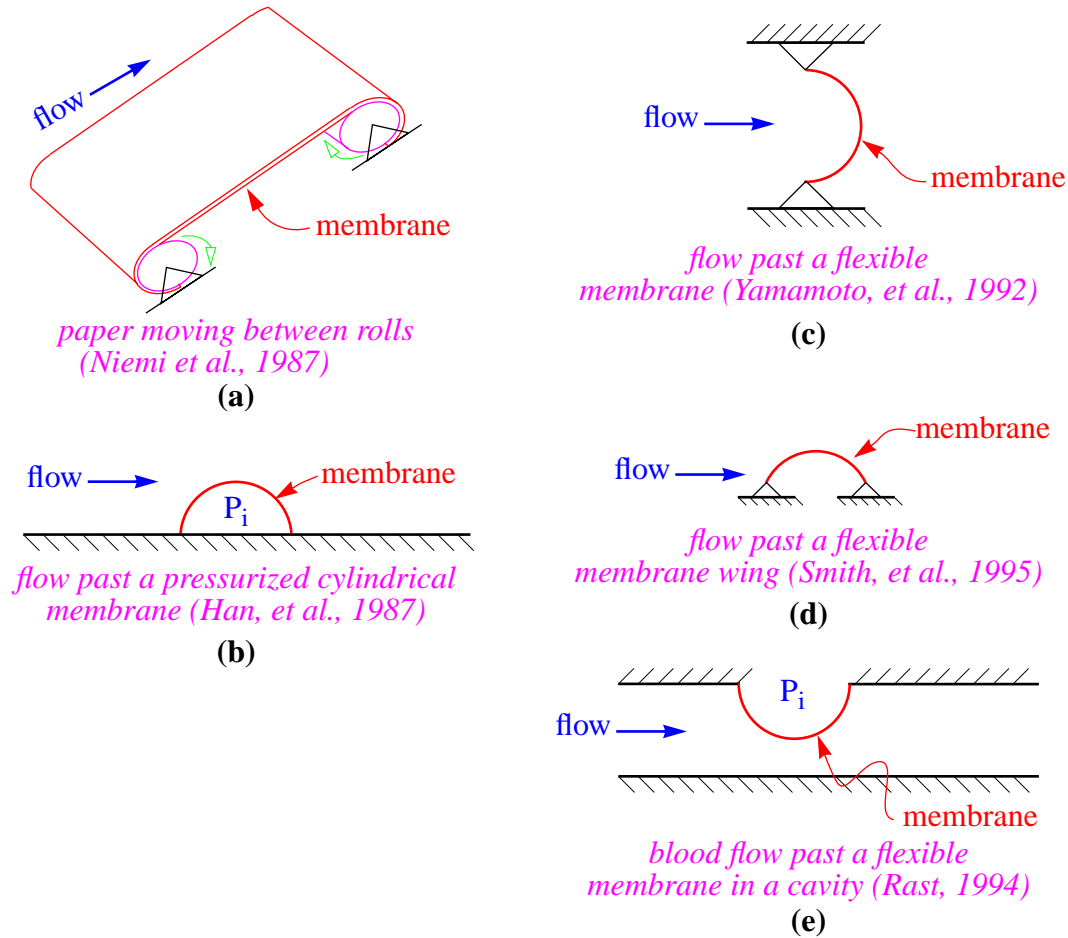


Figure 6. Various Fluid-Membrane Interaction Simulations In The Literature

2.6 Particle-In-Cell Methods

Harlow (1964) reported that the Particle-in-Cell (PIC) method was developed in 1955 at Los Alamos National Laboratory for the solution of complicated fluid dynamics problems. It is a combination of a Lagrangian and an Eulerian method that naturally handles no-slip interfaces between materials and large slippage and distortions. The details of the code are discussed by Amsden (1966).

In general, the idea of the PIC method is to solve the governing equation on an Eulerian grid where derivatives can be conveniently defined. Information is transferred from the grid to Lagrangian material particles via mapping functions. The material particles move or convect and carry with them certain properties. Variations on the method can occur by changing the mapping method. That is, the mapping functions themselves may be changed or the method of mapping may be changed. In Harlow's classical version of PIC, velocities were mapped from the grid to the particle. In a less dissipative version called FLIP (FLuid-Implicit-Particle) (Brackbill, et al. 1986, 1988), material particle velocities are only updated from the grid solution.

An outline of a FLIP-type algorithm is as follows:

- 1.) Solve the governing equation to obtain acceleration at the grid nodes.
- 2.) Integrate the acceleration to obtain the velocity on the grid.
- 3.) Map the acceleration to the particles to update the velocity.
- 4.) Move the particles based on the velocity determined in step 2.
- 5.) Map particle quantities to the grid in preparation for the solution at the next time step,
- 6.) Determine velocity gradient, strains, and stresses at nodes (or vertices),
- 7.) Determine grid forces from stresses.

Sulsky and Brackbill (1991) use a method similar to Peskin's (1977), but based on the PIC method, to simulate suspended bodies moving in a fluid. A force density term, $\mathbf{F}(\mathbf{x}, t)$, is added to equations for Stokes' flow for an incompressible fluid

$$\begin{aligned} -\nabla p + \mu \Delta \mathbf{u} + \mathbf{F} &= 0 \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{2.1}$$

where the gradient ∇ and Laplacian Δ are taken with respect to current position, \mathbf{x} is current position, \mathbf{u} is velocity, p is pressure, and the force, \mathbf{F} , is determined from the sum of internal and external forces. The external forces may be those due to gravity or magnetic fields. The internal forces only exist in the suspended body and are due to the strains within the body. The internal forces are determined from

$$\mathbf{f}^{int}(\mathbf{x}, t) = \nabla \cdot E(\mathbf{x}, t)(\nabla \mathbf{d} + (\nabla \mathbf{d})^T) \quad (2.2)$$

where E is a material modulus that is a function of position because it is zero in the fluid regions, and \mathbf{d} is the displacement field. Peskin's method has also been used for flow where the acceleration terms are significant (Peskin, 1995).

The basic ideas from the PIC or FLIP methods have been adapted recently to solid dynamics by changing step (6) of the FLIP-type algorithm. These field variables are evaluated at material points, and the resulting approach is applied to impact problems with elastic and elastic-plastic constitutive equations (Sulsky et al., 1993; Sulsky et al., 1994).

2.7 Immersed Boundary Methods

There have been several papers published on simulations where a moving boundary or interface is immersed in a fluid or other medium. The common thread to these methods is that they attempt to describe the evolution of the entire system with a common governing equation.

Peskin (1977) studied blood flow through the heart. In his development he attempted to make as little distinction as possible between the fluid and nonfluid (heart muscle) regions. Peskin solved the governing equations on an Eulerian grid. The governing Navier-Stokes equations are modified with a force density term, $\mathbf{F}(\mathbf{x}, t)$, which is nonzero only for the nonfluid regions

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \eta \Delta \mathbf{u} + \mathbf{F} \quad (2.3)$$

where ρ is density, \mathbf{u} is velocity, p is pressure, and η is viscosity.

The heart valves are treated as Lagrangian bars that can only support a tensile or compressive force, and that move within the Eulerian grid. The force in a bar (representing a valve) is a function of the relative displacement of its endpoints. The bars representing the heart muscle are a pair of parallel springs, one in series with an active “contractile element” that causes the springs (muscle) to contract in a prescribed manner. It is the forces in these bars that are interpolated to the Eulerian grid to define $\mathbf{F}(\mathbf{x}, t)$. A more recent paper (Peskin and McQueen, 1995) describes three-dimensional simulations using the immersed boundary method.

2.8 Stability of Fluid-Structure Interaction Formulations

Some information was presented in the above sections regarding stability and time step limits. However, it is felt that this topic warrants its own section as there are several papers where stability is the main subject.

In general, explicit integration schemes lead to a condition on the time step for the calculations to be stable, a situation called conditional stability. In contrast, some implicit schemes are unconditionally stable and the time step can be larger than that of an explicit calculation. The time step for the implicit calculation is determined by the accuracy needed in the solution. Implicit schemes, however, require a system of equations to be solved simultaneously. In certain problems, these equations may be nonlinear, which may increase the solution time. The hope is that the increase in computation time for each step on an implicit scheme can be offset by using significantly larger time steps than allowed by stability constraints of an explicit scheme.

Stability of finite difference and finite element schemes for fluid *or* solid simulations are well documented in the literature. Some new stability issues may arise when fluid and solid simulations are combined.

Tu and Peskin (1992) provide a numerical investigation of stability with their immersed boundary method. The immersed boundary method calculates motion of a Lagrangian structure embedded in a fluid whose governing equations (Stokes flow is assumed) are solved on a regular stationary grid. The effect of the structure on the fluid is determined by interpolating forces from the structure to the grid. The model problem used to examine stability is an elastic closed boundary (like a cylinder) immersed in an incompressible fluid. The closed boundary is perturbed into the form of an ellipse, and then the calculations demonstrate the boundary relaxing into a circular shape.

Tu and Peskin use three methods to calculate the force from the immersed, closed boundary. They are: explicit, approximate implicit, and implicit methods. The approximate implicit method *estimates* the boundary configuration at the end of the time step to calculate the boundary forces. This has been the method used in practice. The stability results are as expected. The most stable method is implicit, followed by approximate implicit, and, finally, explicit. The computational time per time step is ordered in reverse, which is also expected. It is stated that the implicit method is probably too expensive for practical applications.

The work of Neishlos et al. (1981, 1983) on explicit finite difference schemes for fluid-structure simulations shows that the time step limit for the coupled problem may be more severe than that for either the fluid or solid alone for some schemes. It is shown that the reduced limit can be avoided by varying the scheme of differencing the governing equations.

Jones (1981) shows a similar result with Euler-Lagrange fluid-structure coupling. In a one-dimensional spring-piston-fluid coupled problem he indicates that certain “injudicious choices” of the coupling formulation and discretization parameters can make an unconditionally stable Implicit Continuous-fluid Eulerian (ICE) algorithm unstable. He shows this analytically by deriving a damping parameter, D , which has as one of its variables ψ . For $\psi > 0$ the current rather than the advanced structural velocity values are used to update the position of the structure, which is called weak coupling. In the expression for D , this is destabilizing. For strong coupling $\psi = 0$, where advanced velocities are used to move the structure, and the effect is stabilizing. Depending on particular parameters of the problem, such as mass of the piston, spring constant, tube length, coarseness of the discretization, etc., it is possible for D to be negative, and thus result in an unstable algorithm.

Belytschko (1980) describes various methods for integrating the equations for fluid-structure interaction. The following schemes for integrating the equations are discussed: (i) integrating both the solid and fluid equations explicitly with the same and different time steps, (ii) integrating the solid implicitly and fluid explicitly, and (iii) integrating both implicitly. The motivation for the mixed methods is that a single time step for a fluid-structure simulation done purely explicitly may result in unreasonable run times if the solid is stiff compared to the fluid. Also, purely implicit methods may require too much core memory and too many iterations due to large fluid meshes. A detailed comparison is not made between the methods, but several example problems are solved. The conclusion is that the solution method should be picked according to the specific problem.

2.9 Summary and Opportunities for Improvement in Current Approaches

One of the major areas that can be improved upon is the handling of interfaces. Interfaces are the contact points/surfaces between fluid and solids as well as between two or more solids or several fluids. Pure Eulerian methods in general have difficulty accurately simulating boundaries, interfaces, and contact. Lagrangian codes must employ sophisticated contact algorithms to detect and compensate for

interfaces.

Most of the work in fluid-structure simulations uses coupled codes, including ALE schemes. With coupling, there may be two separate codes that are coupled or a single code that uses an Eulerian fluid description and a Lagrangian solid description. Generally, in ALE simulations the fluid description tends more towards Eulerian and the solid towards Lagrangian. ALE codes still have to employ a general-purpose contact algorithm to handle generation of new contact surfaces. In several published ALE simulations, the solid mesh is initially attached to the Eulerian fluid mesh.

In coupled codes, computational effort must be expended to determine the intersection of the Lagrangian solid with the Eulerian grid. If the “right” mesh velocity is applied to Eulerian fluid elements in an ALE simulation, some of this effort may be avoided. However, determining the optimum mesh velocity in general is not trivial. In addition, interface equations that prescribe continuity of normal velocity may have to be solved (ref Fig. 4).

The material point method offers potential advantages if applied to fluid-structure interaction problems. One of these advantages stems from the fact that the material point method uses both an Eulerian and Lagrangian mesh description. There is better resolution of material boundaries and mixed cells and a natural way of handling history-dependent materials as compared to purely Eulerian methods. Highly distorted flow can be simulated without causing mesh distortion, which is a common problem in updated Lagrangian methods. A no slip contact algorithm is automatic to the method. That is, it comes at no additional computational expense.

The material point method can use a common momentum equation for both the compressible fluid and the solid. The fluid is to be treated no differently than the solid except in the constitutive routine to determine stress as a function of strain rate. Thus, the enforcement of potentially nonlinear boundary conditions between the fluid and solid is not an issue. The method will automatically enforce continuity of normal velocities between the fluid and the solid.

The material point method developed for history-dependent materials is described in Chapter 3. Modifications to the MPM to handle compressible, inviscid or viscous fluids and membranes are discussed in subsequent chapters.

CHAPTER 3. THE MATERIAL POINT METHOD

The MPM evolved from a particle-in-cell (PIC) method called FLIP (FLuid-Implicit-Particle) (Brackbill and Ruppel, 1986), which itself is an improved version of Harlow's (1964) original PIC method. Several references were made in Section 2.6 to descriptions of the PIC, FLIP, and MPM methods. In the extension of FLIP to history-dependent materials, strain and stress are now defined on the material points as opposed to the cell centers. This is the significant difference between FLIP and the MPM.

To begin the description of the MPM, we start with the equation for conservation of linear momentum given by

$$(\rho \boldsymbol{\sigma}^s) \cdot \nabla + \rho \mathbf{b} = \rho \mathbf{a} . \quad (3.1)$$

The specific stress, $\boldsymbol{\sigma}^s$, is defined as $\boldsymbol{\sigma}/\rho$ and is symmetric. The constitutive equation is given, for the moment, in rate form by

$$\dot{\boldsymbol{\sigma}}^s = \mathbf{T}^s : \dot{\boldsymbol{\epsilon}} \quad (3.2)$$

where the strain rate is the symmetric part of the velocity gradient

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}[(\mathbf{v}\nabla) + (\mathbf{v}\nabla)^T] . \quad (3.3)$$

The gradient operator $(\)\nabla$ is with respect to the current configuration. The specific body force is \mathbf{b} , and the acceleration is \mathbf{a} . The strain rate, $\dot{\boldsymbol{\epsilon}}$, is defined to be the symmetric part of the gradient of velocity, \mathbf{v} , while \mathbf{T}^s is the tangent modulus tensor relating specific stress rate and strain rate.

To obtain weak forms of the governing equations, suppose equations 3.1 through 3.3 are multiplied by the weighting functions, \mathbf{W} , $\rho \mathbf{w}$, and $\rho \mathbf{w}^*$, in turn, and an integration over the current configuration, Ω , is performed. After the use of the divergence theorem for the first equation (3.1), the resulting set of equations is

$$\int_{\Omega} \rho [\mathbf{W} \cdot \mathbf{a} + \boldsymbol{\sigma}^s : \mathbf{W}\nabla] d\mathbf{v} = \int_{\Omega} \rho \mathbf{W} \cdot \mathbf{B} d\mathbf{v} + \int_{\partial\Omega} \mathbf{W} \cdot \boldsymbol{\tau} ds \quad (3.4)$$

$$\int_{\Omega} \rho \mathbf{w} : [\dot{\boldsymbol{\sigma}}^s - \mathbf{T}^s : \dot{\boldsymbol{\epsilon}}] d\mathbf{v} = 0 \quad (3.5)$$

$$\int_{\Omega} \rho \mathbf{w}^* : \left\{ \boldsymbol{\varepsilon} - \frac{1}{2} [(\mathbf{v} \nabla) + (\mathbf{v} \nabla)^T] \right\} dv = 0 \quad (3.6)$$

Here, $\boldsymbol{\tau}$ denotes the prescribed part of the traction on the surface $\partial\Omega$. Differentials of volume and surface are denoted by dv and ds , respectively.

Now approximate the variables in equations 3.4, 3.5, and 3.6 over a discrete domain using shape or basis functions. The objective is to obtain a set of equations that can be solved on a discrete domain to approximate the solution to the continuous form of the governing equations (3.1 through 3.3).

First, consider a mesh of material points with Ω_p denoting a subdomain of Lagrangian material points. Associated with each subdomain is a material point \mathbf{X}_p^o in the reference configuration and a material point \mathbf{X}_p in the current configuration. At $t=0$ the configurations are Ω_p^o as shown in Fig. 7. The choice of precisely where the material point is located in a particular subdomain is arbitrary.

At a later time the original configurations deform into configurations Ω_p . The deformation is arbitrary, and the material points are tracked with their reference vectors \mathbf{X}_p . The boundary material points and interior points are treated the same.

Over the material points, define piecewise constant basis functions U_p such that $U_p = 1$ for all material points on Ω_p ; otherwise $U_p = 0$. It follows that $U_i(\mathbf{X}_j) = \delta_{ij}$. Suppose these basis functions are used to represent the functions \mathbf{w} , \mathbf{w}^* , $\boldsymbol{\sigma}^s$ and $\boldsymbol{\varepsilon}$ in the weak form of equations 3.5 and 3.6. For example,

$$\boldsymbol{\sigma}^s = \sum_{p=1}^N \boldsymbol{\sigma}_p^s U_p \quad \boldsymbol{\varepsilon} = \sum_{p=1}^N \boldsymbol{\varepsilon}_p U_p \quad (3.7)$$

where $\boldsymbol{\sigma}_p^s$ and $\boldsymbol{\varepsilon}_p$ are the time-dependent stress and strain variables which are taken to be constant over Ω_p .

Dirac delta functions are used as basis functions for the mass density:

$$\rho = \sum_{p=1}^N M_p \delta[\mathbf{X} - \mathbf{X}_p] \quad (3.8)$$

where M_p is the mass of particle p .

Then, with the argument that the generalized variables \mathbf{w}_p and \mathbf{w}_p^* are arbitrary, equations 3.5 and 3.6

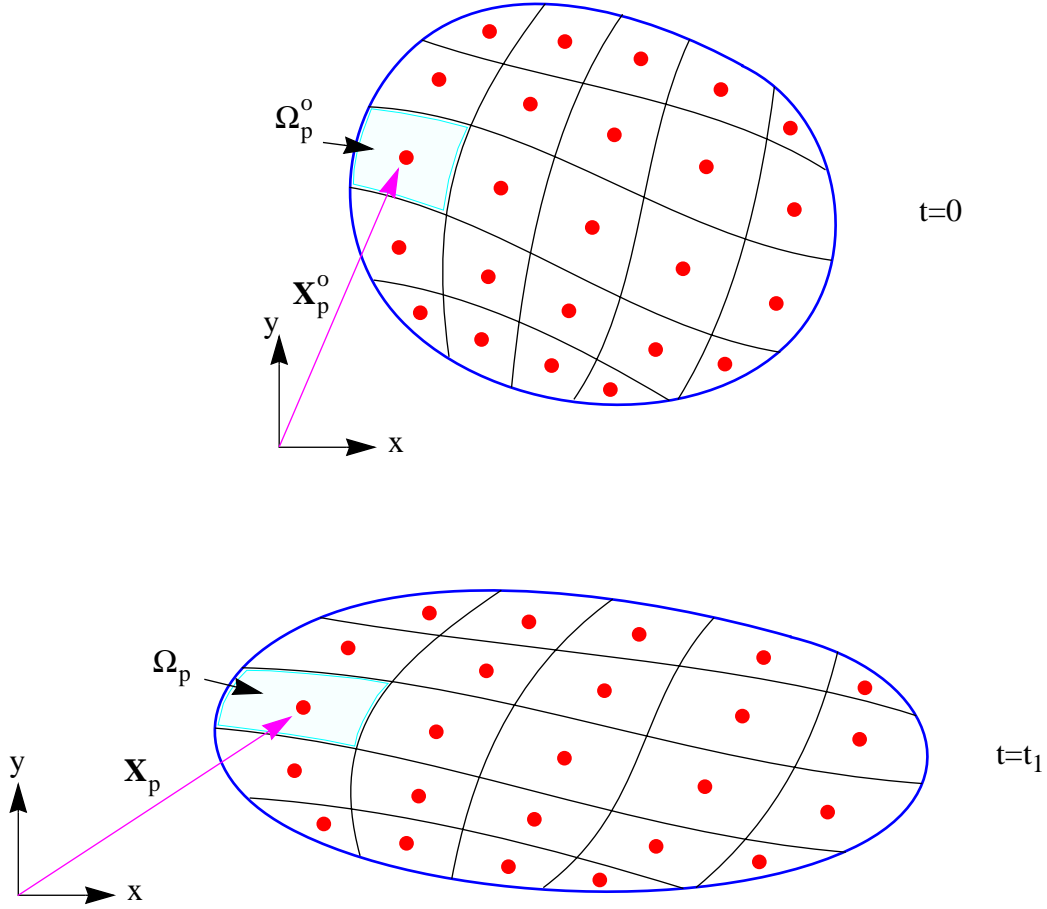


Figure 7. Mesh of Lagrangian Material Points and Their Subdomains

become:

$$\dot{\boldsymbol{\sigma}}_p^s = \mathbf{T}^s(\mathbf{X}_p) : \dot{\boldsymbol{\epsilon}}_p \quad (3.9)$$

and

$$\dot{\boldsymbol{\epsilon}}_p = \frac{1}{2}[(\mathbf{v}\nabla) + (\mathbf{v}\nabla)^T]_{\mathbf{X}_p} \quad (3.10)$$

in which the subscript, \mathbf{X}_p , in the expression for the strain rate indicates that the function in the square brackets is evaluated at the material point. In other words, the stress and strain rate are evaluated at those material points which will be tracked as part of the computational procedure. This is the principal difference between the MPM and other approaches such as the finite element method and the original FLIP algorithm. In many finite element codes, stress and strain rates are determined at element centers or at Gauss points. The original FLIP algorithm for fluids applies constitutive equations ele-

ment-by-element to compute the stress at element centers. A similar approach has also been used for linear elastic solids where there is no history dependence in the constitutive equations. By contrast, the MPM makes greater use of the material points to maintain current values of history variables and material parameters; and constitutive equations are evaluated at material points rather than at element centers.

Now, define a mesh over the spatial computational domain, and let \mathbf{x} define position in this domain. The grid nodes in the computational domain are labeled \mathbf{x}_i . This spatial mesh can be thought of as conventional finite elements and will be used in addition to the mesh of material points. Figure 8 illustrates a mesh of material points overlaid on the Eulerian computational mesh with spatial nodes \mathbf{x}_i . Suppose these elements are used to form nodal basis functions $N_i(\mathbf{x})$ associated with spatial

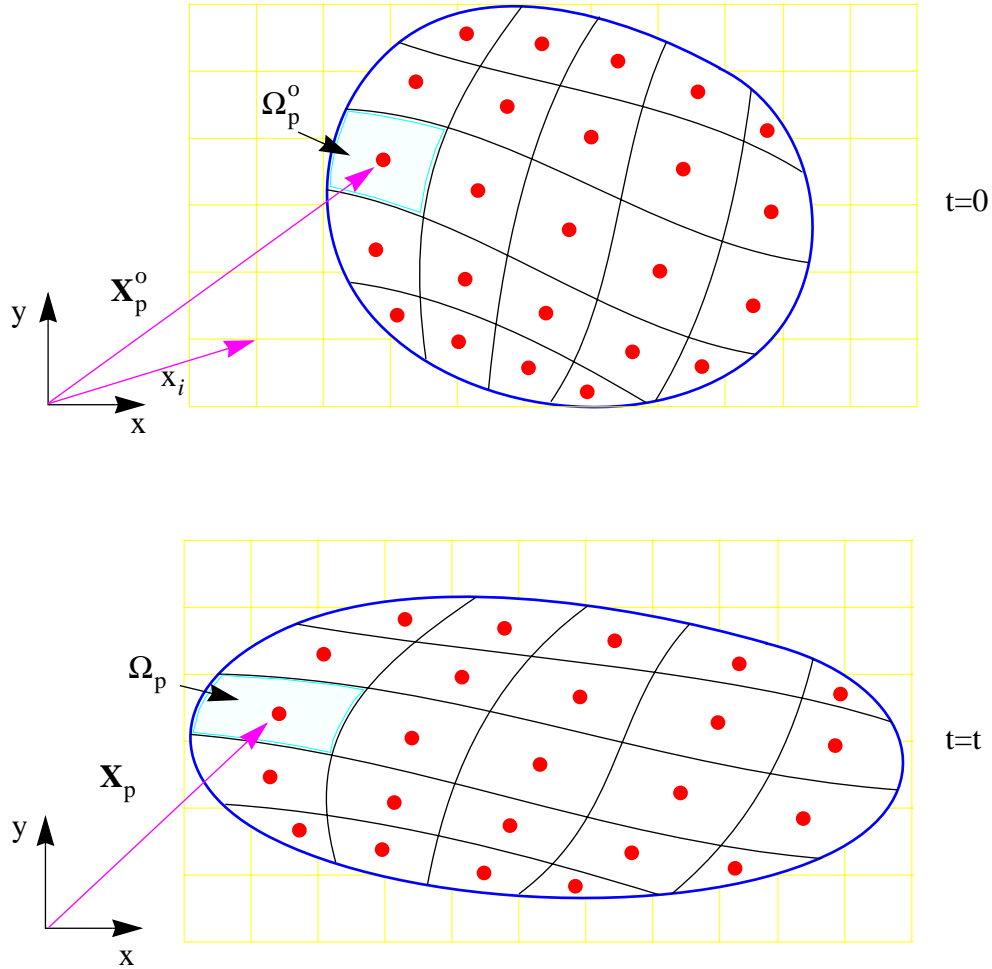


Figure 8. Mesh of Lagrangian Material Points Overlaid on the Computational Mesh

points \mathbf{x}_i with $i=1, \dots, n$. The remaining variables \mathbf{W} , \mathbf{v} and \mathbf{a} must be continuous at least in the limit as the spatial mesh size goes to zero. The conventional finite element representations for the continuous variables are:

$$\mathbf{W} = \sum_{i=1}^n \mathbf{W}_i N_i(\mathbf{x}) \quad (3.11)$$

$$\mathbf{v} = \sum_{i=1}^n \mathbf{v}_i(t) N_i(\mathbf{x}) \quad (3.12)$$

$$\mathbf{a} = \sum_{i=1}^n \mathbf{a}_i(t) N_i(\mathbf{x}) \quad (3.13)$$

in which \mathbf{W}_i , \mathbf{v}_i and \mathbf{a}_i denote the nodal vectors for the respective functions. Introduce the mapping matrix, $[\mathbf{S}]$, whose components, S_{pi} , are values of the nodal basis functions at the current locations of the material points. Also define the set of gradient vectors, \mathbf{G}_{pi} , which represent the gradient of each basis function at the current locations of the material points. The components of these matrices are

$$S_{pi} = N_i(\mathbf{X}_p) \quad \text{and} \quad \mathbf{G}_{pi} = N_i \nabla|_{\mathbf{X}_p}. \quad (3.14)$$

Consider the weak form of the equation of motion given as equation 3.4. With the use of equations 3.8, and 3.11 through 3.14, the first and second terms in the equation become:

$$\int_{\Omega} \rho \mathbf{W} \cdot \mathbf{a} dv = \sum_{i,j=1}^n \mathbf{W}_i m_{ij} \cdot \mathbf{a}_j \quad \text{where} \quad m_{ij} = \sum_{p=1}^N \mathbf{M}_p S_{ip}^T S_{pj} \quad (3.15)$$

$$\int_{\Omega} \rho \boldsymbol{\sigma}^s : \mathbf{W} \nabla dv = \sum_{i=1}^n \mathbf{W}_i \cdot \mathbf{F}_i^{\text{int}} \quad \text{where} \quad \mathbf{F}_i^{\text{int}} = - \sum_{p=1}^N \mathbf{G}_{ip}^T \mathbf{M}_p \boldsymbol{\sigma}_p^s \quad (3.16)$$

The term m_{ij} denotes a component of the mass matrix associated with the computational grid, and $\mathbf{F}_i^{\text{int}}$ is the internal force vector associated with node i . Nodal vectors for the body force field, \mathbf{b}_i , and the surface traction are defined as a natural consequence of the volume and surface integrals involving applied forces:

$$\int_{\Omega} \rho \mathbf{W} \cdot \mathbf{b} dv = \sum_{i=1}^n \mathbf{W}_i \cdot \mathbf{b}_i \quad \text{where} \quad \mathbf{b}_i = \sum_{p=1}^N S_{ip}^T \mathbf{M}_p \mathbf{b}_p \quad (3.17)$$

$$\int_{\Omega} \mathbf{W} \cdot \boldsymbol{\tau} dv = \sum_{i=1}^n \mathbf{W}_i \cdot \hat{\boldsymbol{\tau}}_i \quad \text{where} \quad \hat{\boldsymbol{\tau}}_i = \int_{\partial\Omega} \mathbf{N}_i \boldsymbol{\tau} ds \quad (3.18)$$

Alternatively, since the body force and surface traction are explicitly given, the conventional finite element form can be used in which these functions are evaluated at the grid nodes. The external force vector, $\mathbf{F}_i^{\text{ext}}$, is obtained from the body force and surface traction in a consistent manner. The components of \mathbf{W}_i are arbitrary except for those points where components of the displacement are prescribed. With the understanding that the constraints on the displacement field are invoked, the weak form of the equation of motion yields

$$\sum_{j=1}^n m_{ij} \mathbf{a}_j = \mathbf{F}_i^{\text{int}} + \mathbf{F}_i^{\text{ext}} \quad i = 1, \dots, n \quad (3.19)$$

The set of equations given by 3.9, 3.10 and 3.19 are similar in form to those obtained by conventional finite element schemes. For example, the internal and external force vectors associated with nodes are developed by sweeping over elements at each time step. The components of physical vectors are arranged sequentially to form a vector of scalar components. The constitutive equation sub-routines are also traditional.

Equation 3.19 is to be solved at discrete times, t^k . After the acceleration is obtained at time level k , the grid node velocity is advanced over the time interval $\Delta t = t^{k+1} - t^k$ using this acceleration. A simple explicit integration gives

$$\tilde{\mathbf{v}}_i^{k+1} = \mathbf{v}_i^k + \Delta t \mathbf{a}_i^k. \quad (3.20)$$

The tilde on $\tilde{\mathbf{v}}_i^{k+1}$ indicates that these are temporary grid velocities that will be replaced later by mapping from material points.

After equation 3.19 is solved for grid acceleration, the material point velocities, positions, strains, and stresses are updated to reflect this new solution. This part of the solution procedure assumes an updated Lagrangian frame. That is, the grid has moved in this time increment Δt with each node having a velocity as defined in equation 3.20. Figure 9 illustrates this concept.

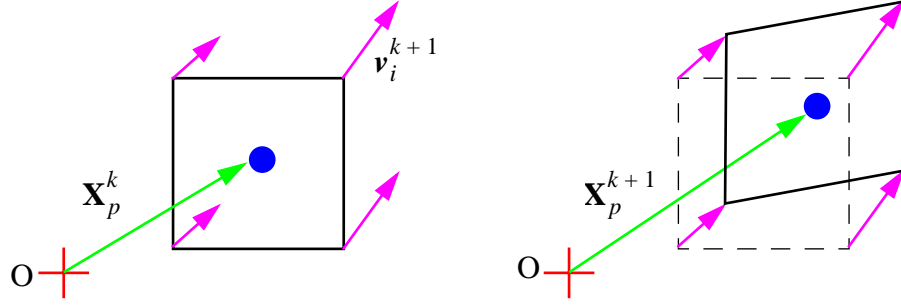


Figure 9. Material Point Convection

The new material point positions, determined by moving them in the computed velocity field given by equation 3.12, are

$$\mathbf{X}_p^{k+1} = \mathbf{X}_p^k + \Delta t \sum_{i=1}^n \tilde{\mathbf{v}}_i^{k+1} \mathbf{N}_i(\mathbf{X}_p^k) \quad (3.21)$$

and since the grid uses isoparametric elements the shape function values at the new time level are the same as the previous time level. The velocity field in which the material points move is single valued so that unphysical interpenetration of two materials does not occur.

The material point velocities are also updated according to

$$\mathbf{v}_p^{k+1} = \mathbf{v}_p^k + \Delta t \sum_{i=1}^n \mathbf{a}_i^k \mathbf{N}_i(\mathbf{X}_p^k) . \quad (3.22)$$

The strain increment at material points is updated using the new grid velocity values and discrete gradient as

$$\Delta \boldsymbol{\epsilon}_p = \frac{\Delta t}{2} \sum_{i=1}^n \left\{ \mathbf{G}_{pi}^k \tilde{\mathbf{v}}_i^{k+1} + (\mathbf{G}_{pi}^k \tilde{\mathbf{v}}_i^{k+1})^T \right\} \quad (3.23)$$

where \mathbf{G}_{pi} are defined in equation 3.14. With the updated material point velocities now considered known, “convected” nodal quantities \mathbf{v}_j^{k+1} (note lack of tilde) are obtained using a least squares weighting of material point velocities

$$\sum_{j=1}^n m_{ij}^{k+1} \mathbf{v}_j^{k+1} = \sum_{p=1}^N m_p \mathbf{v}_p^{k+1} \mathbf{N}_i(\mathbf{X}_p^{k+1}) . \quad (3.24)$$

Next we shall consider the conservation of mass. The mass of each material point is constant in the MPM. Thus, conservation of mass is satisfied if the mapping of mass from the material points to the grid is conservative. The grid mass is given in terms of the material point mass (from equation 3.15) as

$$m_{ij} = \sum_{p=1}^N M_p S_{ip}^T S_{pj} . \quad (3.25)$$

It can be shown in general that the sum of all entries of m_{ij} equals the sum of M_p , which is due to the sum over grid points of the shape functions being unity. For example, consider a single material point in a cell as shown in Fig.

10. The 4x4 mass matrix is given by

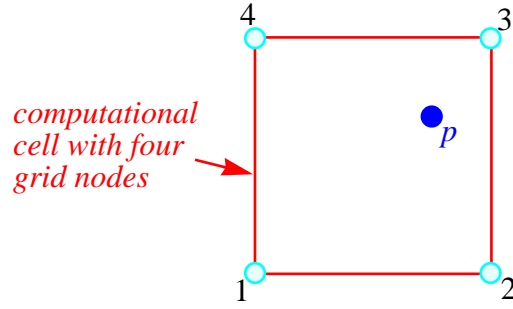


Figure 10. A Material Point in a Computational Cell

$$m_{ij} = M_p \begin{Bmatrix} S_{11} \\ S_{12} \\ S_{13} \\ S_{14} \end{Bmatrix} \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \end{bmatrix} = \begin{bmatrix} S_{11}M_p S_{11} & S_{11}M_p S_{12} & S_{11}M_p S_{13} & S_{11}M_p S_{14} \\ S_{12}M_p S_{11} & S_{12}M_p S_{12} & S_{12}M_p S_{13} & S_{12}M_p S_{14} \\ S_{13}M_p S_{11} & S_{13}M_p S_{12} & S_{13}M_p S_{13} & S_{13}M_p S_{14} \\ S_{14}M_p S_{11} & S_{14}M_p S_{12} & S_{14}M_p S_{13} & S_{14}M_p S_{14} \end{bmatrix} . \quad (3.26)$$

If the rows are summed to obtain the diagonal form, and the common term $\sum_{i=1}^4 S_{1i}$ is factored from

each diagonal element, the expression simplifies to

$$\sum_i m_{ij} = \sum_{i=1}^4 S_{1i} \begin{bmatrix} S_{11}M_p & 0 & 0 & 0 \\ 0 & S_{12}M_p & 0 & 0 \\ 0 & 0 & S_{13}M_p & 0 \\ 0 & 0 & 0 & S_{14}M_p \end{bmatrix} = \begin{bmatrix} S_{11}M_p & 0 & 0 & 0 \\ 0 & S_{12}M_p & 0 & 0 \\ 0 & 0 & S_{13}M_p & 0 \\ 0 & 0 & 0 & S_{14}M_p \end{bmatrix} \quad (3.27)$$

where it is seen that the sum of all four diagonal terms equals M_p since $\sum_{i=1}^4 S_{1i} = 1$. In general

$$\sum_j m_{ij} \equiv m_i = \sum_j \sum_p M_p S_{ip}^T S_{pj} = \sum_p M_p S_{ip}^T \quad (3.28)$$

and

$$\sum_i m_i = \sum_i \sum_p M_p S_{ip}^T = \sum_p M_p. \quad (3.29)$$

Therefore, the grid mass is equal to the material point mass. In practice the diagonal form of the mass matrix is used in equation 3.19 to avoid inverting the mass matrix (or solving a set of equations simultaneously); this is common practice in explicit finite element schemes used for structural dynamics simulations.

The final conservation law considered is conservation of energy. For simulation of solid dynamics, the current formulation of the MPM is isothermal. That is, the energy equation is not solved. However, the use of an equation of state for solids that may depend on temperature is not precluded and could be implemented if desired.

The MPM computational algorithm is summarized as follows:

- 1) initialize material point locations, velocities, strains, and stresses,
- 2) map momentum, mass, and internal forces to the grid,
- 3) solve the momentum equation on the grid,
- 4) update material point locations based on grid velocity and update their velocities,
- 5) obtain convected grid velocities,
- 6) determine new velocity gradients and strains and stresses at the material points,
- 7) regrid,
- 8) go to 2.

Momentum is used in step 4 to prevent numerical problems due to division in some cases by a small grid mass value. Grid mass is determined by mapping material point masses to the grid with the shape functions. Thus, the small grid mass is due to the small value(s) of shape functions. When momentum is used, the numerator and denominator of the material point equations are balanced by the shape function value. That is, the numerator and denominator both contain a multiplication by the shape function, and numerical problems are avoided.

The equations below summarize the algorithm for the first time step for a linear elastic material in one dimension. Time levels are denoted by superscripts; “0” is the first time step, and “L1” is the first Lagrangian step. The equations marked as “*” and “**” below are the implementation of the momen-

tum formulation; note the shape function $N_i(x_p^{L1})$ in the numerator of these equations and mass in the denominator.

$$m_i^0 = \sum_p M_p N_i(x_p^0)$$

2) Map material point mass and momentum to the grid:

$$v_i^0 m_i^0 = \sum_p M_p v_p^0 N_i(x_p^0)$$

3) Solve the momentum equation on the grid:

$$m_i^0 (v_i^{L1} - v_i^0) / \Delta t = f_i^0$$

$$\text{let } p_i = m_i^0 v_i^{L1}$$

4) Update material point velocities and locations:

$$v_p^{L1} = v_p^0 + \sum_i \Delta t f_i^0 N_i(x_p^{L1}) / m_i^{L1} *$$

$$x_p^{L1} = x_p^0 + \sum_i \Delta t p_i N_i(x_p^{L1}) / m_i^0 **$$

5) Obtain convected grid velocities: $\tilde{v}_i^{L1} = \sum_p M_p v_p^{L1} N_i(x_p^{L1}) / m_i^{L1}$

6) Determine velocity gradients, strains and stresses at material points where the strain increments, Δe_p^{L1} , are based on the symmetric part of the velocity gradient, E is Young's modulus, and σ_p^{L1} is the

$$\Delta e_p^{L1} = \Delta t \sum_j G_{jp}^{L1} \tilde{v}_j^{L1}$$

new stress: (note: this is for one dimension)

$$\sigma_p^{L1} = E \Delta e_p^{L1}$$

7) Regrid: The grid can be regenerated as specified by the user. In this work the same grid is used throughout the computations to simplify the algorithm.

CHAPTER 4. THE MPM FOR FLUIDS AND VALIDATION OF THE FLUID FORMULATION

Here we give the specific constitutive equation and energy equation used to simulate problems involving compressible fluids. The general development of the momentum equation in Section 3 continues to hold except that the stress tensor for a fluid point (as opposed to a solid point) is given as

$$\sigma_{ij} = 2\mu\dot{\epsilon}_{ij} - \frac{2}{3}\mu\dot{\epsilon}_{kk}\delta_{ij} - p\delta_{ij} \quad (4.1)$$

where μ is shear viscosity, and p is pressure which is usually determined from an equation-of-state. Equation 4.1 assumes the Stokes condition $\lambda = -(2/3)\mu$ where λ is bulk viscosity, and as a result, the static pressure (spherical component of stress) is equal to the thermodynamic pressure. Due to the presence of the delta function, only the diagonal terms in the stress tensor contain $\dot{\epsilon}_{kk}$ and pressure terms. That is, when $i \neq j$, $\delta_{ij} = 0$.

Equation 4.1 is applied point-by-point. The strain rates for a material point are given in two dimensions by

$$\dot{\epsilon}_p \equiv \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} \quad (4.2)$$

and are approximated with the discrete gradient operator as

$$\dot{\epsilon}_p = \frac{1}{2} \sum_{i=1}^N \left\{ \mathbf{G}_{pi}^k \mathbf{v}_i^{k+1} + (\mathbf{G}_{pi}^k \mathbf{v}_i^{k+1})^T \right\}. \quad (4.3)$$

The equation of state implemented is for an ideal gas where pressure is related to density and internal energy by

$$p = (\gamma - 1)\rho i \quad (4.4)$$

where i is specific internal energy, and γ is the ratio of specific heats. Equation 4.4 is applied point-by-point.

There are several methods of determining material point density for use in calculating pressure in

equation 4.4. The first is *cell density* where all material points in a cell are assigned the cell density. That is

$$\rho_c = \sum_{p \in c} m_p / V_c \quad \text{and} \quad \rho_p = \rho_c \text{ for } p \in c \quad (4.5)$$

where m_p is the mass of material point p , and V_c is the volume of cell c . *Quadratic interpolation* determines the cell density as

$$\rho_c = \sum_p m_p S_{cp}^{(2)} / V_c \quad (4.6)$$

where $S_{cp}^{(2)}$ is a quadratic interpolation function whose support in one-dimension spans one cell to the left and right of the cell containing material point p (ref Appendix 3.2, page 174). Material point density is determined the same as in equation 4.5. The *continuity equation* can also be used to update each material point's density as

$$\rho_p^{k+1} = \frac{\rho_p^k}{(1 + \Delta t (\nabla \cdot \mathbf{v})_p)} \quad (4.7)$$

where $(\nabla \cdot \mathbf{v})_p$ is the divergence of velocity evaluated at the material point.

The equation of state (4.4) is dependent upon internal energy as well as density. Thus, the energy equation must be solved for each fluid point. Conservation of energy requires that

$$\rho \frac{di}{dt} = \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} \quad (4.8)$$

where i is the specific internal energy, $\boldsymbol{\sigma}$ is the symmetric stress tensor defined in equation equation (4.1), and $\dot{\boldsymbol{\epsilon}}$ is the strain rate. Using equations 4.1 and 4.2 this expression can be expanded (for two-dimensional Cartesian system with x and y coordinates) in terms of velocity and its derivatives as

$$\rho \frac{di}{dt} = 2\mu \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right) + \left(-\frac{2}{3}\mu \right) (\nabla \cdot \mathbf{v})^2 - p(\nabla \cdot \mathbf{v}) + \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 \quad (4.9)$$

where μ is shear viscosity, p is pressure, and u and v are the x and y components of velocity. This equation is solved using the velocity derivatives that are also used to calculate material point strains. Once the derivatives are calculated it is straight-forward to update the energy according to equation

4.9. The energy update for a fluid point is

$$\dot{i}^{k+1} = \dot{i}^k + \frac{(t^{k+1} - t^k)}{\rho_p^k} \left[2\mu \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right) + \left(-\frac{2}{3}\mu \right) (\nabla \cdot \mathbf{v})^2 - p(\nabla \cdot \mathbf{v}) + \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 \right] \quad (4.10)$$

which is calculated for each fluid material point every time step.

It is well known that most numerical simulations of compressible-fluid shocks provide more accurate results if some type of artificial viscosity is used at the shock front. The artificial viscosity implemented in the MPM is similar to that described by Wilkins (1980). An additional term, q , is added to the material point pressure under shock conditions as

$$q = (\sqrt{c_{max}^2 g}) \rho \tilde{\lambda} D \left[c_1 + \frac{c_2 |\nabla \cdot \mathbf{v}|}{\max |\nabla \cdot \mathbf{v}|} \right] \quad (4.11)$$

where

$$D = \begin{cases} \nabla \cdot \mathbf{v} & \nabla \cdot \mathbf{v} < 0 \\ 0 & \nabla \cdot \mathbf{v} > 0 \end{cases} \quad (4.12)$$

and c_{max}^2 is the maximum sound speed in the fluid, g is a geometric constant proportional to the mesh size, $\tilde{\lambda}$ is an artificial bulk modulus, ρ is density, and c_1 and c_2 are constants. Variable D defined in equation 4.12 forces the artificial viscosity, q , to be zero unless the material point is in compression.

The MPM computational algorithm with the modifications for fluid simulation is summarized as follows:

- 1) initialize material point locations, velocities, strains, and stresses,
- 2) map momentum, mass, and internal forces to the grid,
- 3) solve the momentum equation on the grid,
- 4) update material point locations based on grid velocity and update their velocities,
- 5) obtain convected grid velocities,
- 6) determine new velocity gradients and strains and stresses at the material points; if the material point is fluid then (i) apply the fluid constitutive equation and (ii) update the fluid point's energy,
- 7) regrid,
- 8) go to 2.

4.1 Simulation of Shock Propagation in a Fluid (Sod's Problem)

The objective of the simulation is to test the implementation of the MPM fluid formulation with a problem that has an analytical solution to determine if it is a viable method for future fluid-structure interaction simulations. The MPM results for pressure, density, velocity, and energy can be compared to their theoretical values.

Sod (1978) investigated finite difference schemes for simulation of a shock propagating through fluids. His model problem consists of a shock tube where a diaphragm separates two regions which have different densities and pressures. Initially the regions have zero velocity. At time $t=0$, the diaphragm is broken. Figure 11 illustrates the initial conditions of the problem. The shock strength is defined as the ratio of p_2/p_1 . For this problem the shock strength is approximately 3.0, which gives a shock speed of about 1.75 (Liepman, 1957). The shock is allowed to travel a distance of 0.25, which it should do in 0.143 seconds.

This simulation is done with the formulations previously described. This is a one-dimensional problem, however, it is solved with the MPM in two-dimensions, and the solution variables are con-

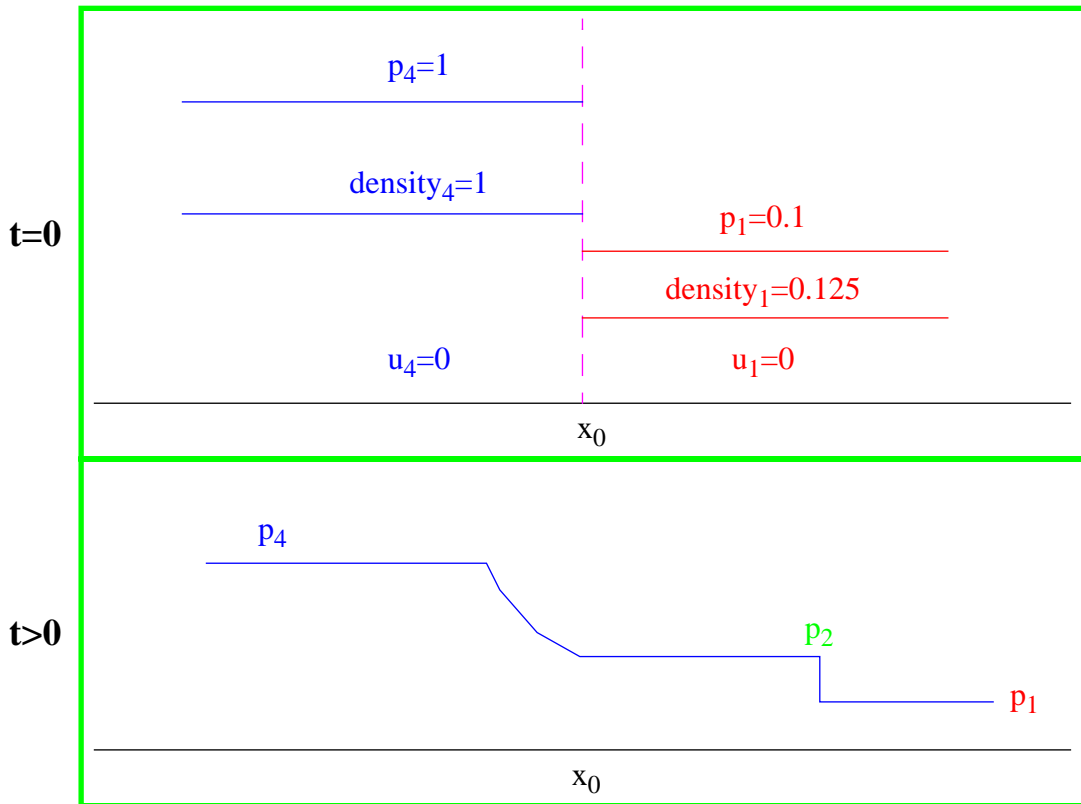


Figure 11. Sod's Fluid Shock Propagation Problem

stant with respect to the y (vertical) direction.

The initial set-up for the simulation is shown in Fig. 12. The square grid used is 100×1 with a cell dimension of 0.01, and nine material points are initially placed in each cell.

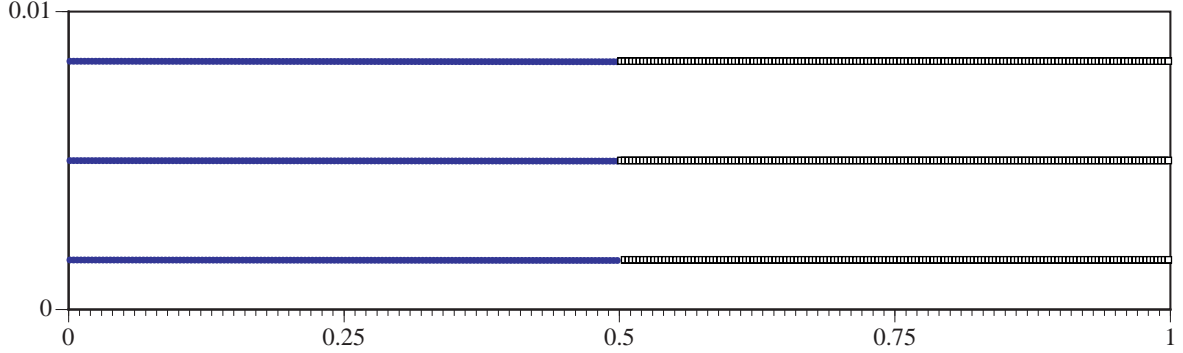


Figure 12. Initial material point Positions for the MPM Simulation of Sod's Problem

It was observed during the first attempts at using the MPM for simulation of Sod's problem that the results appeared noisy. It was noted that in FLIP some smoothing of the data is accomplished using higher order interpolation. Thus, several methods of data smoothing were investigated in the MPM to improve the solution to Sod's problem. The main contributor to the noise seemed to be the material point density calculation. Figure 13 shows the density at cell centers for the three methods described previously (page 33), with the solid line being the theoretical value. The plot marked *cell density* is a result of calculating the density of a cell and assigning all material points in that cell the same density. The plot marked *quadratic interpolation* smooths the densities over multiple cells, and the plot marked *continuity equation* uses the continuity equation to update each material point's density.

It is obvious that either the *quadratic interpolation* or the *continuity equation* improves the results significantly over the *cell density* method. The results for the *continuity* and *quadratic* calculations are very similar. When corresponding data points in the two plots are subtracted, the maximum difference is less than 0.005 which is small compared to density values ranging between 0.125 and 1.0.

The results of the MPM simulations using quadratic density calculations are shown in Figure 14 with circles marking each data point, and the theoretical values are shown with a solid black line. In Fig. 14(a), top, the simulation is performed without artificial viscosity, and the bottom figure shows how the artificial viscosity smooths the oscillations at the shock front. All data are calculated and plot-

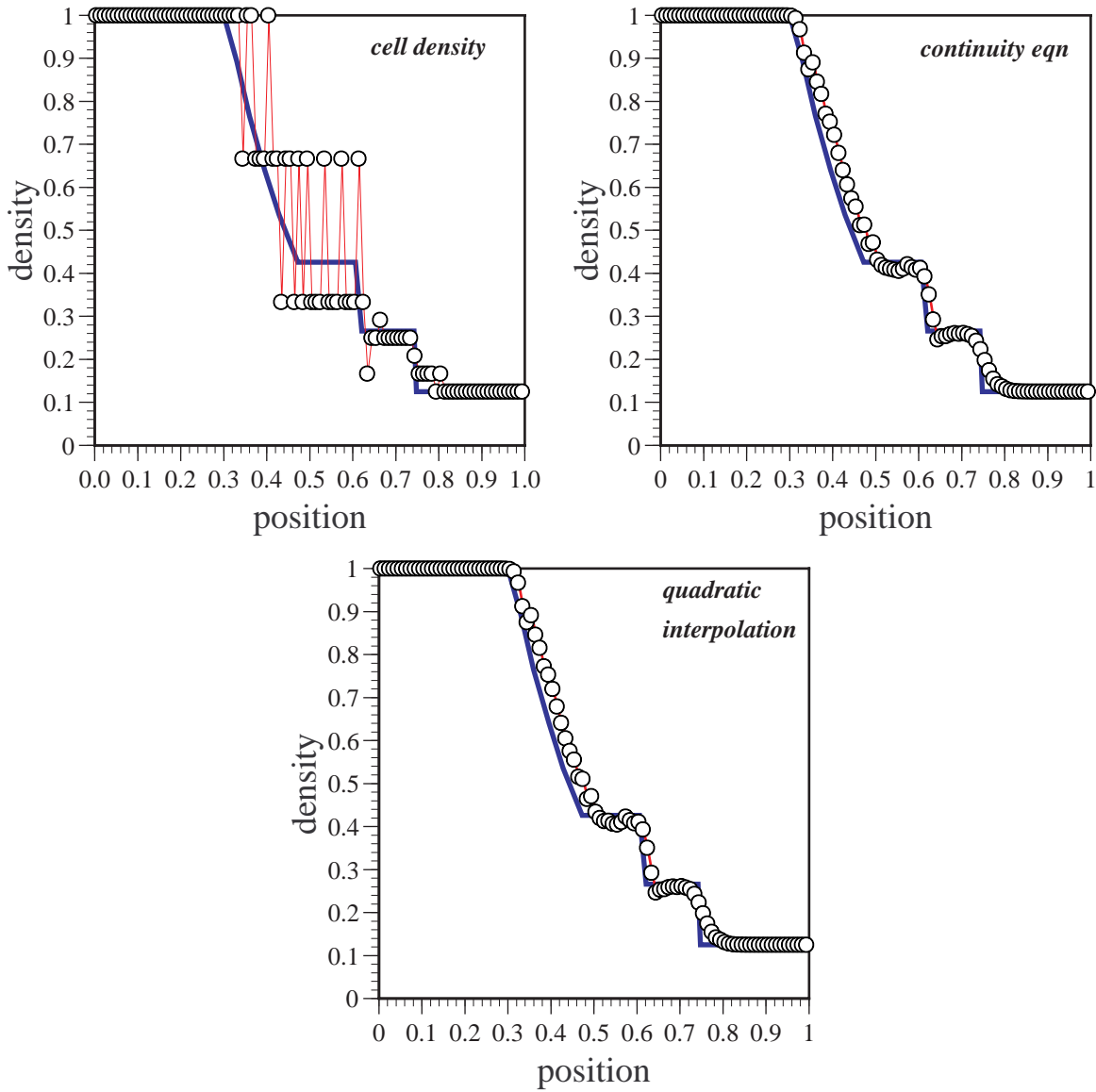
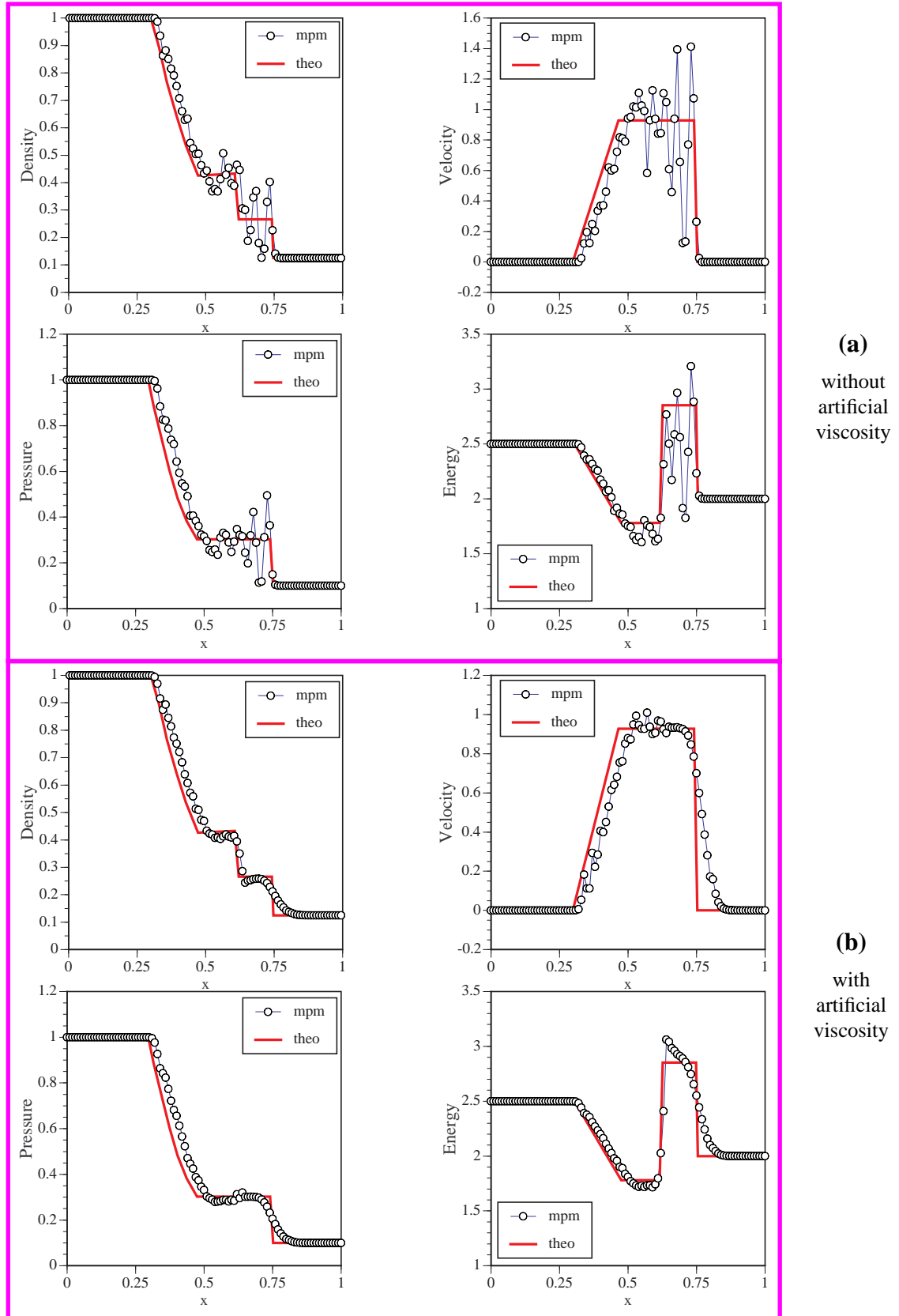


Figure 13. Comparison of Density Calculations in Sod's Problem

ted at grid vertices except density which is at cell centers.

The results show reasonable agreement with theory. A qualitative comparison of this method with the twelve methods compared in Sod's paper indicates that the results here are better than the worst methods but not as accurate as the best methods. Since the objective of this work is to investigate fluid-structure interaction, no attempt was made to optimize the MPM for simulation of compressible fluids. The results were accurate enough to show the viability of using the MPM for simulating fluid-structure interaction.



4.2 Gas Expansion

This simulation was run solely as another test of the MPM fluid formulation. The physical problem consists of allowing an initially cylindrical body of gas of radius 0.38 to expand into a vacuum. The MPM simulation is compared to the same simulation run implicitly and explicitly with FLIP.

Figure 15 shows the problem set up which was run with 1/4 symmetry conditions on the x and y axes. The gas is considered to be inviscid.

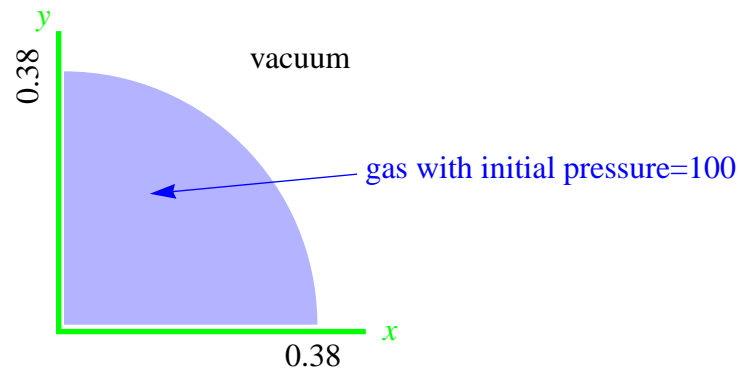


Figure 15. Gas Expansion Problem

The mesh size used is 0.025×0.025 with nine material points per cell. Figure 16 shows the calculated material point positions for the three simulations at $t=0.005$. The material points are overlaid on the 20×20 grid.

The MPM simulation most closely agrees with the implicit FLIP simulation. The scattering and non-physical bunching of material points in the explicit FLIP calculation is attributed to the difficulties that this formulation has with the large pressure gradient at the edge of the expanding gas. Figures 17 and 18 show the results for pressure and specific internal energy. The MPM shows reasonable agreement with the implicit FLIP calculation.

4.3 Chapter 4 Summary

Simulations have been performed with model fluid problems with the standard MPM formulation subject to certain fluid-specific modifications for density and shock effects. The results compare well with finite-difference methods and with the implicit FLIP version. Existing references show that the MPM is a viable method for solid continua. This chapter shows that the same basic formulation is

equally suitable for both solid and fluid continua.

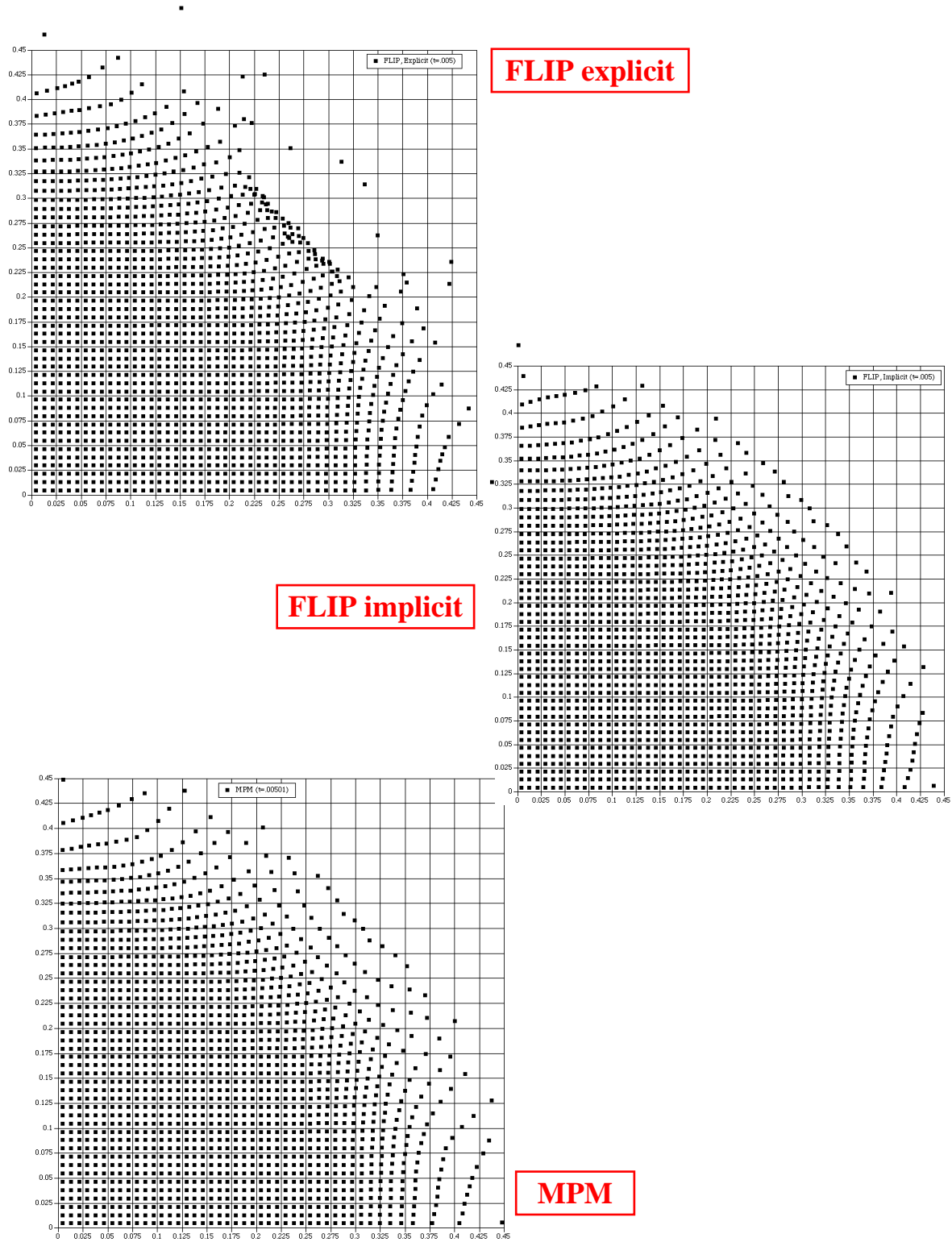


Figure 16. Gas Expansion Results from the MPM and FLIP-Material Point Positions

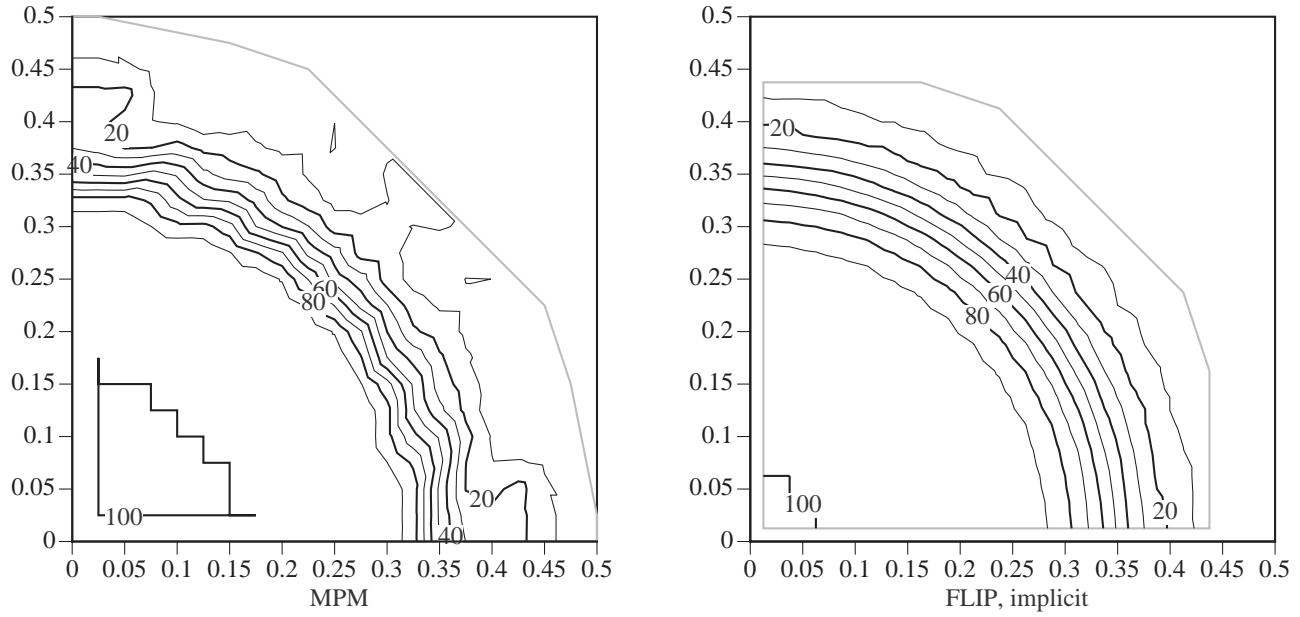


Figure 17. Gas Expansion Pressure Contours from the MPM and FLIP

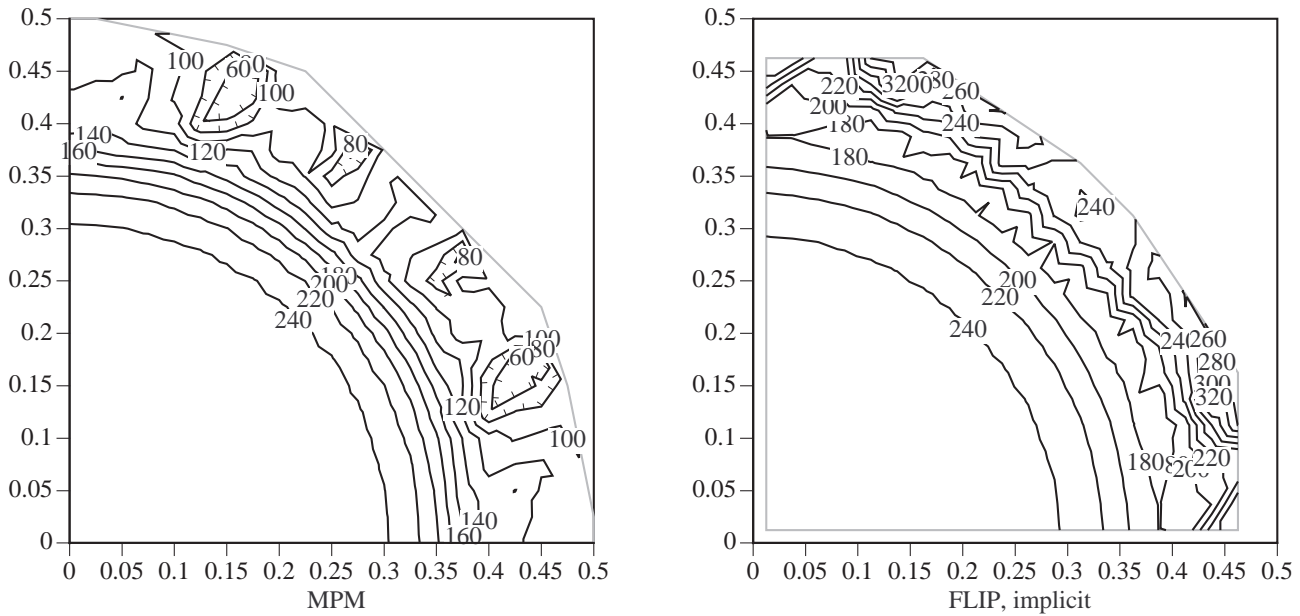


Figure 18. Gas Expansion Specific Internal Energy from the MPM and FLIP

CHAPTER 5. THE MPM FOR MEMBRANES AND VALIDATION OF THE MEMBRANE FORMULATION

A membrane is a thin-walled structure that has stiffness only in the plane tangent to the structure and ideally no stiffness in bending. The stress components and traction, \mathbf{t} (vector of units force per unit area), through the thickness of a membrane are constant. Figure 19 shows an arbitrarily shaped membrane supporting an external force \mathbf{F} . On the right is shown a slice of the membrane in the x - y

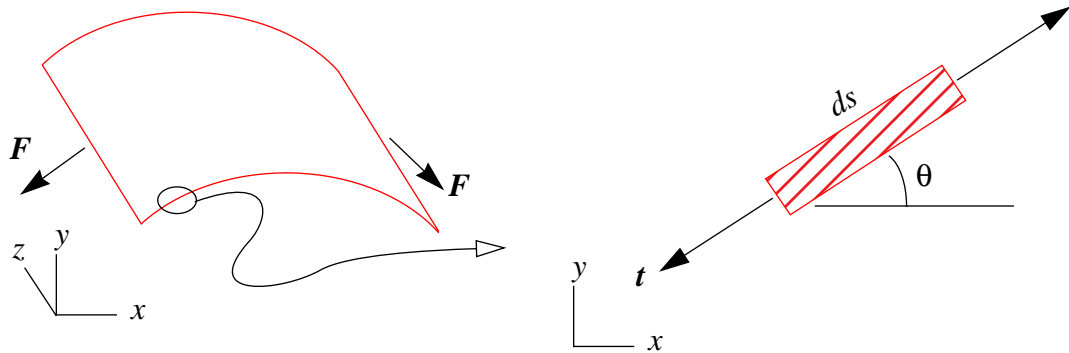


Figure 19. Three-Dimensional Representation of a Membrane (left) and Section View

plane of width ds inclined at angle θ . The traction on the edges of the membrane does not vary through the thickness, and the traction acts in the plane of the membrane or tangent to the membrane, in this case oriented at angle θ to the x - y plane.

There are no physical structures that exhibit perfectly ideal membrane behavior. However, there are many examples of materials that can be approximated using membrane theory. Some examples of these materials are thin-walled metallic structures, thin-walled elastomeric materials such as balloons or diaphragms, paper, most fabrics, and many types of biological entities such as the walls of cells or simple organisms. Strings and thin wire can be considered a one-dimensional membrane.

A string or wire can only support a tangent stress which is due to a tangent strain. The strains in the other two perpendicular directions are such that the stresses in these directions are zero. This is a uniaxial stress formulation. A membrane surface is similar except that stresses exist only in the plane of the surface, i.e. a state of plane stress is assumed. The out-of-plane strain components must be adjusted to provide the plane stress formulation. Furthermore, in either case, the tangent stress does not vary through the thickness.

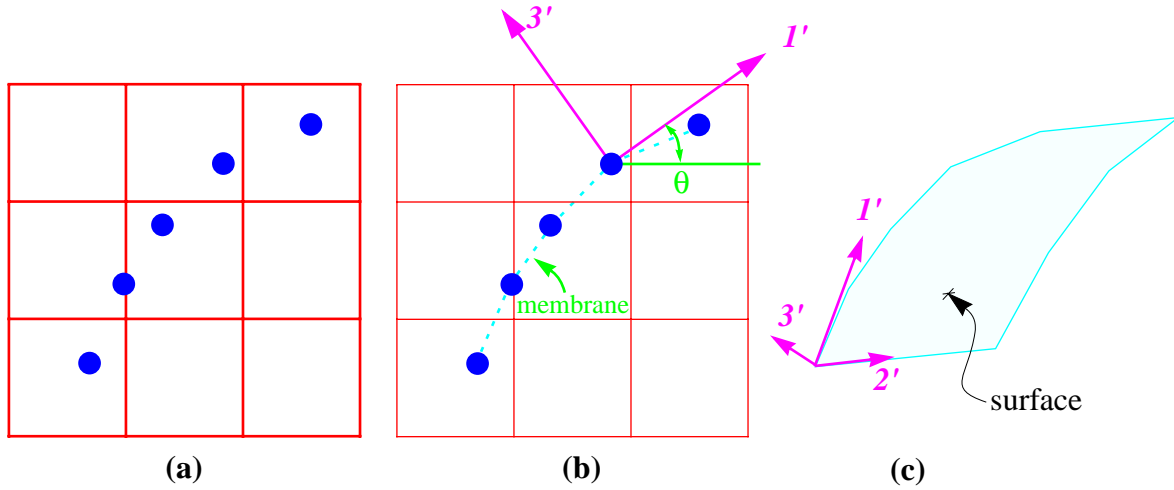


Figure 20. (a) Points in MPM Simulation, (b) Physical Representation of Membrane and its Local Coordinate System and (c) Perspective View of Membrane Surface

The initial question concerned how to implement the MPM for membrane problems. Several variations of multiple material points through the thickness were investigated. Here we show that a particularly simple version of only a single layer of material points is quite adequate. One of the key points is to determine the strains in the local normal-tangential coordinate system and adjust them to be consistent with the membrane assumptions.

Figures 20(a) and 20(b) illustrate a configuration of material points that might represent a string or membrane. Figure 20(a) shows the material points in the computational grid. Figure 20(b) shows the interpretation of the string or membrane (dashed line) represented by the material points and the local tangential-normal ($1'$ - $3'$) coordinate system at a representative material point. If a uniaxial stress formulation is used, the string in Fig. 20(b) has stress only in the tangential ($1'$) direction. If a plane stress formulation is used, this same arrangement of material points may also represent a membrane as shown in Fig. 20(c). Here, the stress is in the $1'$ - $2'$ surface. The implementation of the string and membrane formulations is described in the following sections.

5.1 Uniaxial Stress Formulation for a Spring or String

This section describes the formulation of a method for simulation of an isotropic elastic spring (bar) or string.

The strain rate is defined by the symmetric part of the velocity gradient as

$$\dot{\mathbf{\epsilon}} = \frac{1}{2}(\mathbf{v}\nabla + (\mathbf{v}\nabla)^T) \quad (5.1)$$

where \mathbf{v} is the velocity of material point p . The total calculated material point strain at time level $k+1$ is $\tilde{\mathbf{\epsilon}} = \mathbf{\epsilon}_t^k + \dot{\mathbf{\epsilon}}\Delta t$ where $\mathbf{\epsilon}_t^k$ is the total tangent strain from the previous time step. These strains are not those used to determine the material point stress, and, thus, use of the tilde. The components of strains $\tilde{\mathbf{\epsilon}}$ in the tangential-normal (I' - $3'$) coordinate system are calculated. To be consistent with the physics of a string, the $3'$ and $2'$ (into the page in Fig. 20) strains are adjusted so that the only stress is in the tangential (I') direction. The tangent strain increment is given by

$$\Delta\epsilon_t = (\mathbf{t} \cdot \dot{\mathbf{\epsilon}} \cdot \mathbf{t})\Delta t \quad (5.2)$$

where \mathbf{t} is the unit tangent vector directed along the length or surface of the string. The total tangent strain is

$$\epsilon_{I'}^{k+1} = \Delta\epsilon_t + \epsilon_t^k. \quad (5.3)$$

Expanding equation 5.3, the updated tangential strain component is

$$\epsilon_{I'}^{k+1} = \Delta\epsilon_{11}(\cos\theta)^2 + \Delta\epsilon_{22}(\sin\theta)^2 + 2\Delta\epsilon_{12}\cos\theta\sin\theta + \epsilon_t^k \quad (5.4)$$

and the tangent stress is

$$\sigma_{I'}^{k+1} = E\epsilon_{I'}^{k+1} \quad (5.5)$$

where E is Young's modulus. For uniaxial stress the other strains are

$$\epsilon_{2'} = \epsilon_{3'} = -\nu\epsilon_{I'}. \quad (5.6)$$

In practice, the strains in the primed (local) coordinate system (equations 5.2 and 5.6) are input into a matrix multiplication routine to determine the stresses as follows

$$\begin{bmatrix} \sigma_{I'} \\ \sigma_{3'} \\ \sigma_{2'} \\ \sigma_{I'-3'} \end{bmatrix} = [\mathbf{E}] \begin{bmatrix} \epsilon_{I'} \\ \epsilon_{3'} \\ \epsilon_{2'} \\ \epsilon_{I'-3'} \end{bmatrix} = [\mathbf{E}] \begin{bmatrix} \epsilon_{I'} \\ -\nu\epsilon_{I'} \\ -\nu\epsilon_{I'} \\ 0 \end{bmatrix} \quad (5.7)$$

where the elasticity matrix, $[\mathbf{E}]$, is

$$[E] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 \\ \nu & (1-\nu) & \nu & 0 \\ \nu & \nu & (1-\nu) & 0 \\ 0 & 0 & 0 & (1-2\nu) \end{bmatrix}. \quad (5.8)$$

After simplification equation 5.7 reduces to

$$\begin{bmatrix} \sigma_{I'} \\ \sigma_{3'} \\ \sigma_{2'} \\ \sigma_{I'3'} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu)\epsilon_{I'} - 2\nu^2\epsilon_{I'} \\ \nu\epsilon_{I'} - (1-\nu)(\nu\epsilon_{I'}) - \nu^2\epsilon_{I'} \\ \nu\epsilon_{I'} - \nu^2\epsilon_{I'} - (1-\nu)(\nu\epsilon_{I'}) \\ 0 \end{bmatrix} = \begin{bmatrix} E\epsilon_{I'} \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (5.9)$$

After the tangent stress is calculated, these components of stress and strain are transformed back to the global x - y coordinate system to compute the global x and y forces.

5.2 The One-Way Constitutive Equation

An addition can be made to the formulation that enables a material point to accumulate negative strain without causing a compressive stress. The idea is to simulate a wrinkle, which can be thought to occur when a string is in compression, without having to resolve the buckling/bending of the membrane. This enables a more realistic flexible string simulation at low cost. The addition to the method is to have Young's modulus zero for negative tangent strains as shown in Fig. 21.

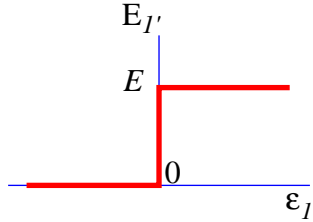


Figure 21. One-Way Constitutive Form to Simulate Wrinkles in a String

5.3 Computational Algorithm

The computational algorithm (as compared to that on [page 31](#)) now has steps 7 and 8 added for

the membrane, and if the “wrinkle” or one-way constitutive model is being used step 7(ii) is followed.

The algorithm now is:

- 1) initialize material point locations, velocities, strains, and stresses,
- 2) map momentum, mass, and internal forces to the grid,
- 3) solve the momentum equation on the grid,
- 4) update material point locations based on grid velocity,
- 5) obtain convected grid velocities,
- 6) determine new velocity gradients and strains and stresses at the material points,
- 7) if the material point is a membrane, determine the tangent vector, rotate total material point strains into the tangent plane, apply the uniaxial or plane stress constitutive model, for either (i) tensile-compressive membranes or (ii) one-way membranes (wrinkle algorithm),
- 8) rotate the stresses back to the global system for evaluation of internal forces,
- 9) regrid,
- 10) go to 2.

5.4 Other Considerations

5.4.1 Resolving the Membrane Forces on a Cartesian Grid

One may ask the question, “Does this even have a chance at working?” It’s a good question when you realize that the membrane forces should be tangent to the membrane in the material point mesh and also on the Cartesian Eulerian mesh, or least the resultant interpolated back to the material points should be. The concept is illustrated in Fig. 22 which shows a line of material points representing a

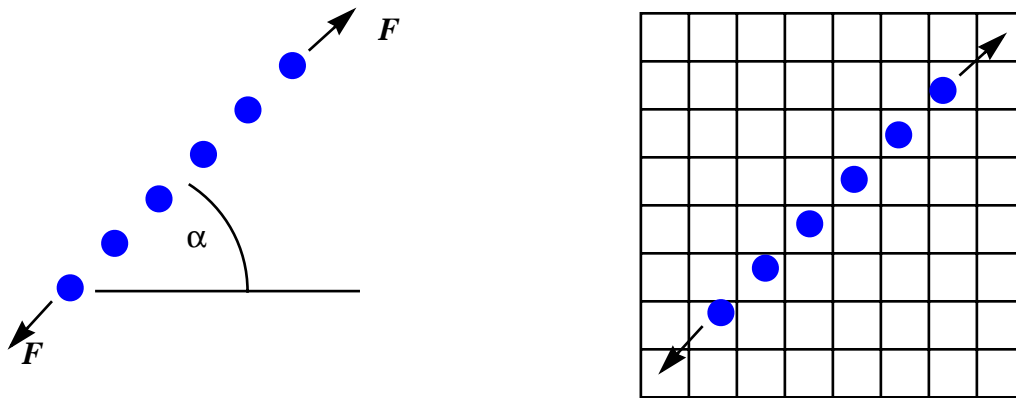


Figure 22. Material Points Representing a Membrane Oriented Obliquely to the Grid

membrane oriented at an arbitrary angle, α , to the horizontal. The picture begs the question, will the mapping of the divergence of the material point stress to the grid result in material point accelerations in a direction tangent to the membrane? One can imagine that if a small enough grid and enough

material points were used that this would be true, but it may not be efficient since the time step is governed by the size of a computational cell. If larger cells are used, what happens when the membrane has curvature inside a single computational cell? The problems solved in this and other sections indicate the MPM membrane formulation does indeed work, and at reasonable grid resolutions.

5.4.2 Rotating Strains

In a prior implementation of step 7 in the computational algorithm of Section 5.3, page 47, only the strain increments were rotated into the plane of the membrane. This was later determined to be incorrect. For large changes in the tangent plane of the membrane during a simulation this resulted in accumulation of stresses that were not tangent to the membrane. Thus, the rotation of these stresses back to the global x - y system could result in forces inconsistent with the membrane orientation. The correct implementation keeps track of the total membrane strains each time step to determine the tangent stress which when rotated back to the x - y coordinate system is consistent with the membrane orientation.

5.4.3 Noise When Material Points Change Cells

Another problem was found when using a small number of material points per cell. In the MPM algorithm the internal force at a grid node is determined by taking the divergence of the stress of each material point that contributes to the grid node. Now consider a 1-D problem where there are only a few material points per cell and one crosses a cell boundary. Since the shape functions are linear, the derivatives of the shape functions which are used to form the divergence are piecewise constant over the elements. The contribution to internal force at a grid node from the material point that just crossed a cell boundary will change sign when the material point crosses the cell boundary. This causes noise in the simulation. In most simulations, the noise is not detectable when examining output. However, in some cases the noise in the material point's velocity may be of the same order as the material point's velocity. This can happen in simulating oscillating problems when a material point's velocity is very small, say at the apex of an oscillation when the velocity goes through zero to change directions. Section A.2, page 171, contains an example to demonstrate this problem.

With this problem noted, several corrections were investigated. One type of solution was aimed at smoothing the stresses. An algorithm to use a cell-centered stress to calculate the internal forces instead of using the stress of each material point was somewhat successful. The addition of more material points also was successful because the noise is overwhelmed by the contribution of internal

forces by other material points. In any event, it was decided not to concentrate on solving this problem at this time since the problem had been identified and “workarounds” were established. There is currently research being conducted using a conjugate gradient method to solve the MPM equations implicitly, and this may have the best potential for a permanent solution to the noise problem.

5.4.4 Setting the Mass of the Membrane Points

The existing algorithm for setting material point mass is cell-based and has been used to initialize masses for material points representing solids. The material points are assigned a fraction of the volume associated with each cell because these material points are initially distributed in a regular pattern in the cells. Figures 23(a) and (b) show the physical solid material and the MPM discretization. The shaded areas represent the volume, Ω_p , associated with each material point which in this case is one-quarter of the cell volume because there are initially four material points per cell. In general, the membrane material points have no ordered relationship with the grid as shown in Figs. 23(c) and (d). Thus, the masses of these material points should be initialized in a different manner.

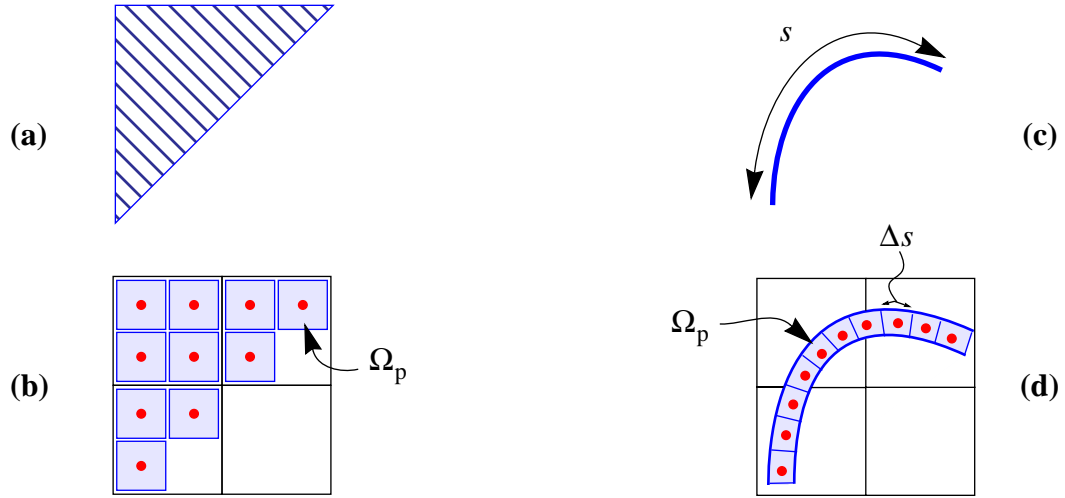


Figure 23. Solid: (a) Physical Representation and (b) MPM Representation and Membrane: (c) Physical Representation (d) MPM Representation

If s is the total length of the membrane, $\hat{\rho}^0$ is the mass per unit area of the membrane material, and N_m is the total number of membrane material points, the mass of a membrane material point can be set as $m_p = s\hat{\rho}^0/N_m$. The quantity Δs , where $\Delta s = s/N_m$, is used to associate a length of the

membrane with each material point. Typically, Δs is a fraction of the smallest cell dimension. Fractions of one-third to one-tenth have been used in simulations. It is essential that the membrane material points not become separated by more than one cell or the membrane will essentially be “broken.”

The effective thickness of the membrane is accounted for in \hat{p}^0 .

The user must beware that if the grid width is different from the membrane thickness then the membrane force will be distributed over a larger area. One can imagine cases where choosing the grid size much larger than the membrane thickness may have unphysical results. This does not seem to have that much of an effect in the cases reported here. The airbag simulations in Section 6.3 use a grid size of 20 mm and a membrane thickness of 0.5 mm, and the results appear physical and show reasonable agreement with experiment.

5.4.5 Multiple Points through the Membrane Thickness

Some work was done in the area of using multiple material points through the thickness of the membrane. This concept is motivated by the idea that you may want to distribute the membrane force over more than one cell width. This idea was quickly abandoned for the reasons outlined below.

Membranes are usually thin, so having multiple material points through the thickness means that the material points have to be very close to one another. To resolve the membrane force over more than one cell width means that the grid in the area of the membrane has to be very small which leads to extremely small time steps. Also, there must be some relationship between the material points through the thickness so that the stress components can be modified to obtain constant stress through the thickness. This requirement leads to the need for connectivity data that can result in a very complicated algorithm.

5.5 Spring-Mass System Simulation

As a first step in validating the membrane formulation, the MPM is used to simulate an oscillating mass on a spring. This problem is simple and has a well defined analytical solution.

5.5.1 Spring-Mass Problem Description

The Spring-Mass problem is illustrated in Fig. 24. A rigid mass is attached to a massless rigid spring of unstretched length L and spring constant k which is connected to a stationary wall. All elements lie in the x - y plane. An acceleration due to gravity, g , acts on the system.

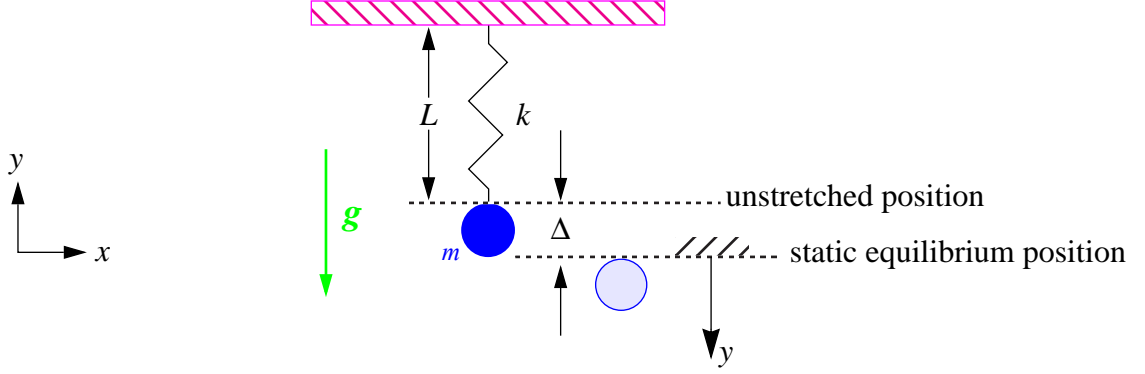


Figure 24. Idealized Spring-Mass System

If position y is measured from the static equilibrium position, the governing differential equation of this system is

$$\frac{\partial^2 y}{\partial t^2} + \frac{k}{m}y = 0 \quad (5.10)$$

The solution to equation 5.10 is

$$y(t) = y_0 \cos \omega t + \frac{\dot{y}_0}{\omega} \sin \omega t \quad (5.11)$$

where $\omega = \sqrt{\frac{k}{m}}$ is the natural frequency, and \dot{y}_0 is the initial velocity. Note that the period of oscillation, T , is given by $T = \frac{2\pi}{\omega}$.

5.5.2 MPM Spring-Mass Vibration Simulation Results

The spring-mass problem is simulated with the MPM method using ten membrane material points. The simulation is approximate because the spring is modeled with non-rigid elastic material points that have mass in contrast with the theoretical problem which has a massless rigid spring. However, the mass of these material points is very small compared to the mass m . The initial position of the material points and the computational grid for the simulation are shown in Fig. 25.

The heavy material point in the simulation has a mass of 3.33 which is 10,000 times the mass of the other material points. The equivalent spring constant, k , of the system is AE/L where A is the cross-sectional area of the spring¹, E is Young's modulus, and L is the length of the string. For this simulation, $A=0.1$, $E=1.0e6$, $g=-250$, and $L=0.3$ which is the distance between the top and bottom

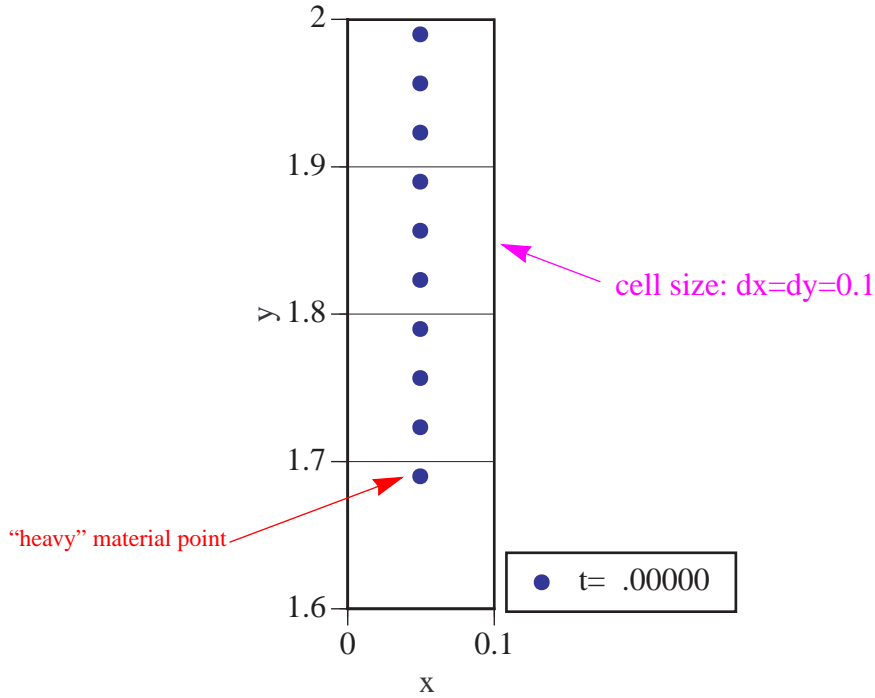


Figure 25. Spring-Mass Vibration Simulation - Initial Positions

material points.

The deflection of the “heavy” material point at the end of the spring is illustrated in Fig. 26 (top). Also plotted in this figure is the deflection of the mass of the equivalent theoretical system which is calculated using the above parameters and equation 5.11. The theoretical kinetic energy as a function of time can be determined by differentiating equation 5.11 to obtain velocity ($KE=mv^2/2$), and the potential energy of the spring is a function of the displacement and spring constant ($PE=kx^2/2$). The potential energy due to gravity (mgh) is not included. The time history of energy in the MPM simulation is shown along with the theoretical calculations in Fig. 26 (bottom).

1. In this context, the spring can be thought of as a bar.

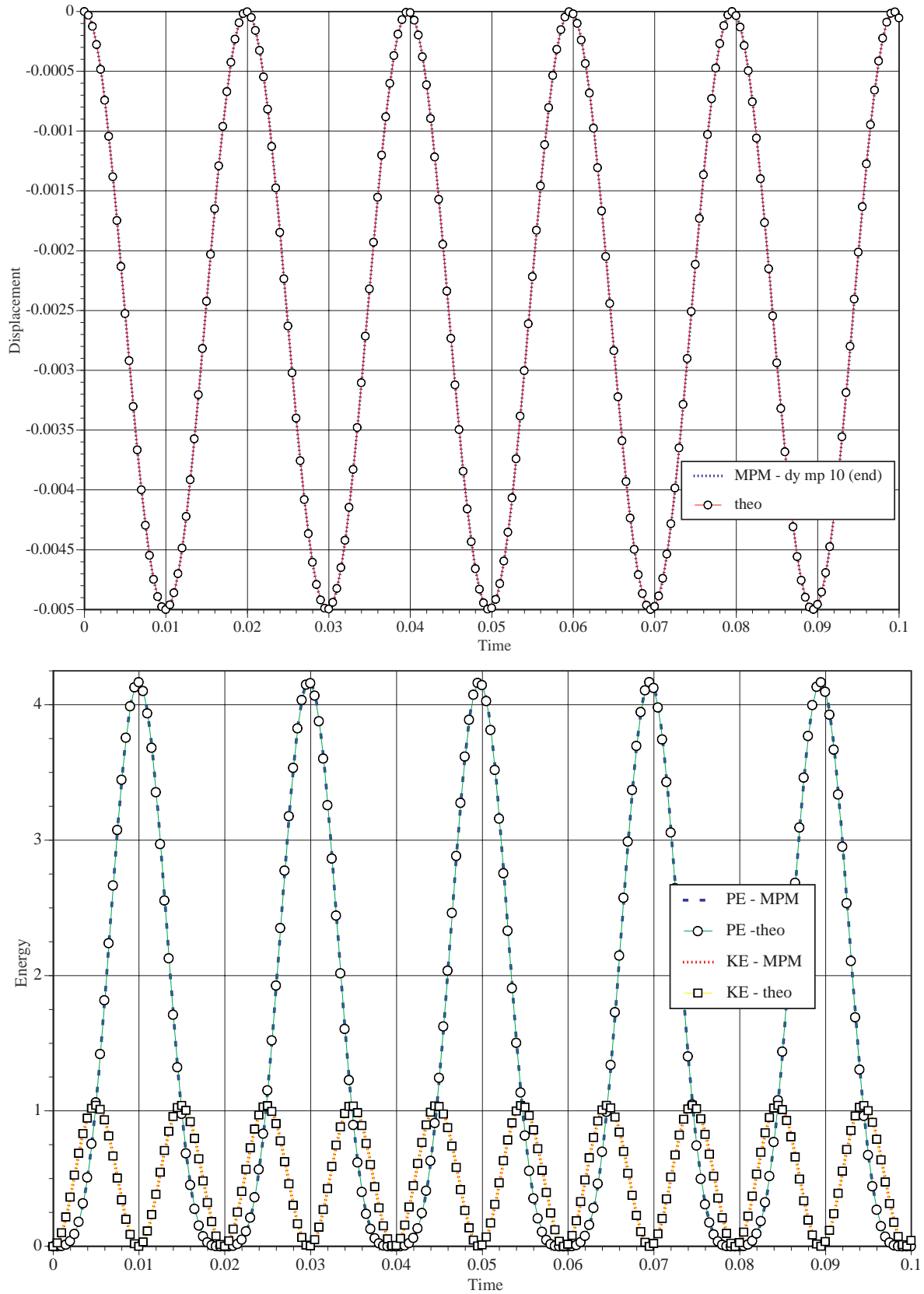


Figure 26. Time History of Mass Displacement (top) and Energy (bottom)

5.6 String-Mass System With Initial Slack

The spring-mass system discussed in the previous section is modified to be a string-mass with initial slack. The physical problem being simulated is illustrated in Fig. 27 along with the MPM representation. Material points 4-9 are given an initial negative strain. With the one-way constitutive model there is no force generated by the material points until the material point has a positive tangent strain (in tension).

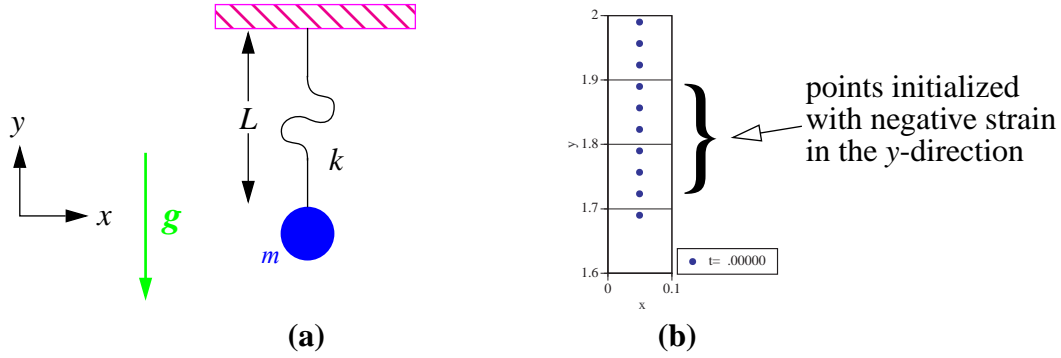


Figure 27. String-Mass with Initial Slack: (a) Physical and (b) MPM Representations

The results are presented in Figures 28 and 29. Figure 28 shows the material point positions at various times. Note that at least one material point is crossing a cell boundary during the oscillation. Figure 29(top) shows the displacement of the bottom material point, and the lower plot is the energy history. This figure warrants some discussion.

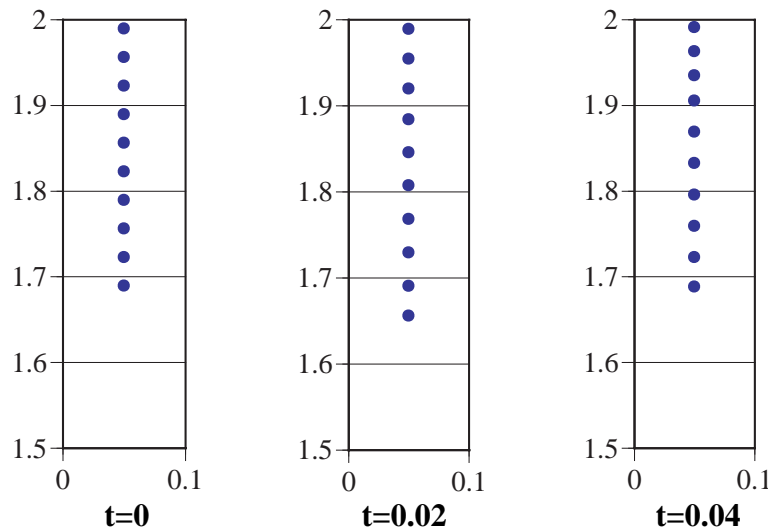


Figure 28. Material Point Positions at Various Times

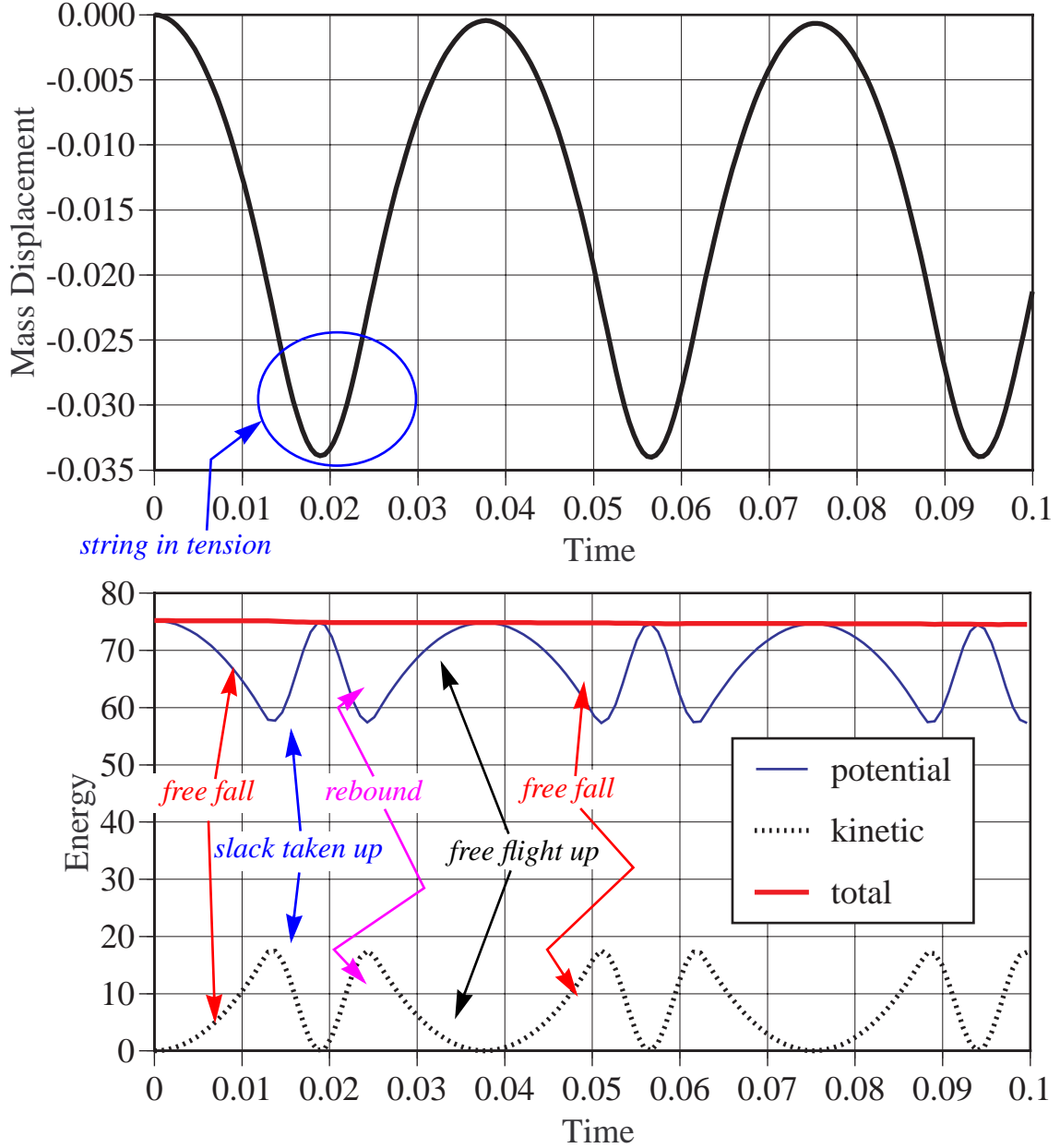


Figure 29. Displacement and Energy Results for the String-Mass Simulation

For $t < 0.013$ the mass is in free fall, and the slack is being taken out of the material points initialized with negative strain. At about $t = 0.013$ the slack is taken up, the string pulls on the mass, and the mass begins to slow down. From here until about $t = 0.023$ the string acts as a spring in tension. Note that the displacement during this time is about 0.01 which is twice that of the mass-spring in Section 5.5 (see Fig. 26, page 53). The mass displaces more here due to its nonzero initial velocity when the string forces begins acting on the mass. At about $t = 0.023$ the string is in compression generating no

resistance force, and the mass begins free flight up until it reaches its apex at about $t=0.0375$. From here the cycle begins again.

5.7 Pendulum Simulation

The pendulum simulation uses the uniaxial stress membrane formulation. This problem demonstrates that the forces due to material point membrane stresses can be adequately resolved on a square Cartesian computation grid. Also, convergence toward an updated Lagrangian finite element solution is demonstrated.

5.7.1 Pendulum Problem Description

The pendulum problem is illustrated in Fig. 30. A mass is attached to a string of length L which is

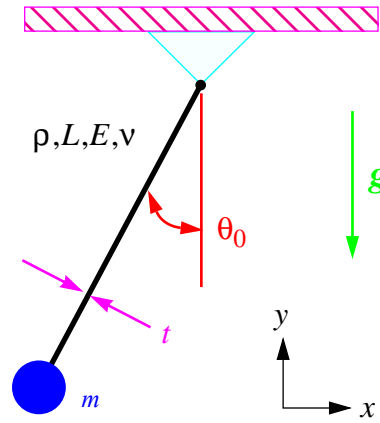


Figure 30. Pendulum Problem Set-Up

connected to a stationary wall. All elements lie in the x - y plane. The string is initially inclined at an angle θ_0 from vertical. A field force due to gravity, g , acts on the system. Table 1 summarizes the problem parameters.

Table 1: Pendulum Simulation Parameters

Item	Value	Item	Value
L	0.73	E	1×10^6
m	3.3	ρ	0.1
g	20.0	ν	0.0
θ_0	19.8°	t	0.1

5.7.2 MPM Pendulum Simulation Results

The pendulum problem is simulated with the MPM and membrane material points. For these simulations, the wrinkle algorithm or one-way constitutive equation described in Section 5.2 is used. Four simulations with different degrees of mesh refinement are presented. An additional simulation performed with an explicit finite element method (FEM) using bar elements is used for comparison purposes. The FEM code used is documented by York (1990).

The initial position of the material points and the computational domain for each simulation is shown in Fig. 31. The edge length of a square computational cell is given by Δ . In each simulation, the material point at the end of the string is given a mass equal to 3.3. Figures 32 through 35 illustrate the position of the material points at various times in the simulations which are carried out for about one and one-half oscillations. The execution time for the simulation with $\Delta=0.1$ on a Macintosh PowerPC 9500 (120 MHz 604 processor) is about 17 minutes and on a IBM-compatible machine with a Pentium II 233 MHz processor is about 9.5 minutes. Note that no attempt has been made to improve the run speed of the code as it is essentially a research code at this time.

Figure 36 shows plots of the FEM simulation. The points (nodes) in Fig. 36 are connected to emphasize this calculation was done with bar elements.

Figure 37 compares the simulations for the angle theta. In the simulations, the angle theta is defined by the angle made by a straight line joining the first and last material points. The MPM simulations agree reasonably well with the explicit FEM simulation, and convergence toward the FEM simulation is observed. As the simulation is refined, the length of the string and direction of stresses in the string are more accurately resolved on the computational grid. The material points stay aligned better as the simulation is refined. The reduced amplitude of oscillations (seen in Figure 37) in the MPM results when compared with the FEM results indicates numerical dissipation. However, the dissipation is decreasing with mesh refinement.

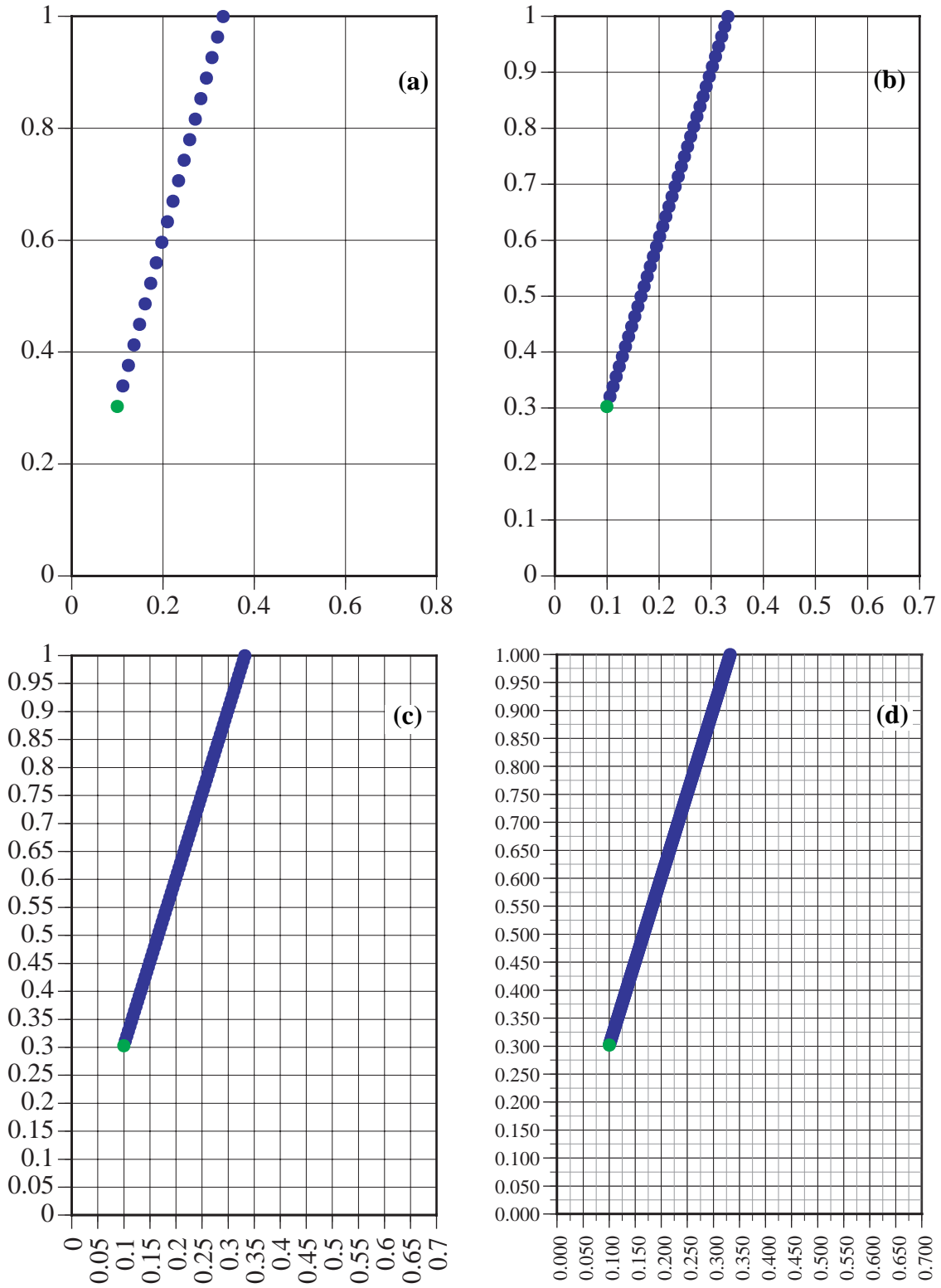


Figure 31. Pendulum Problem Simulations - Initial Conditions for $\Delta =$ (a) 0.2, (b) 0.1, (c) 0.05, and (d) 0.025

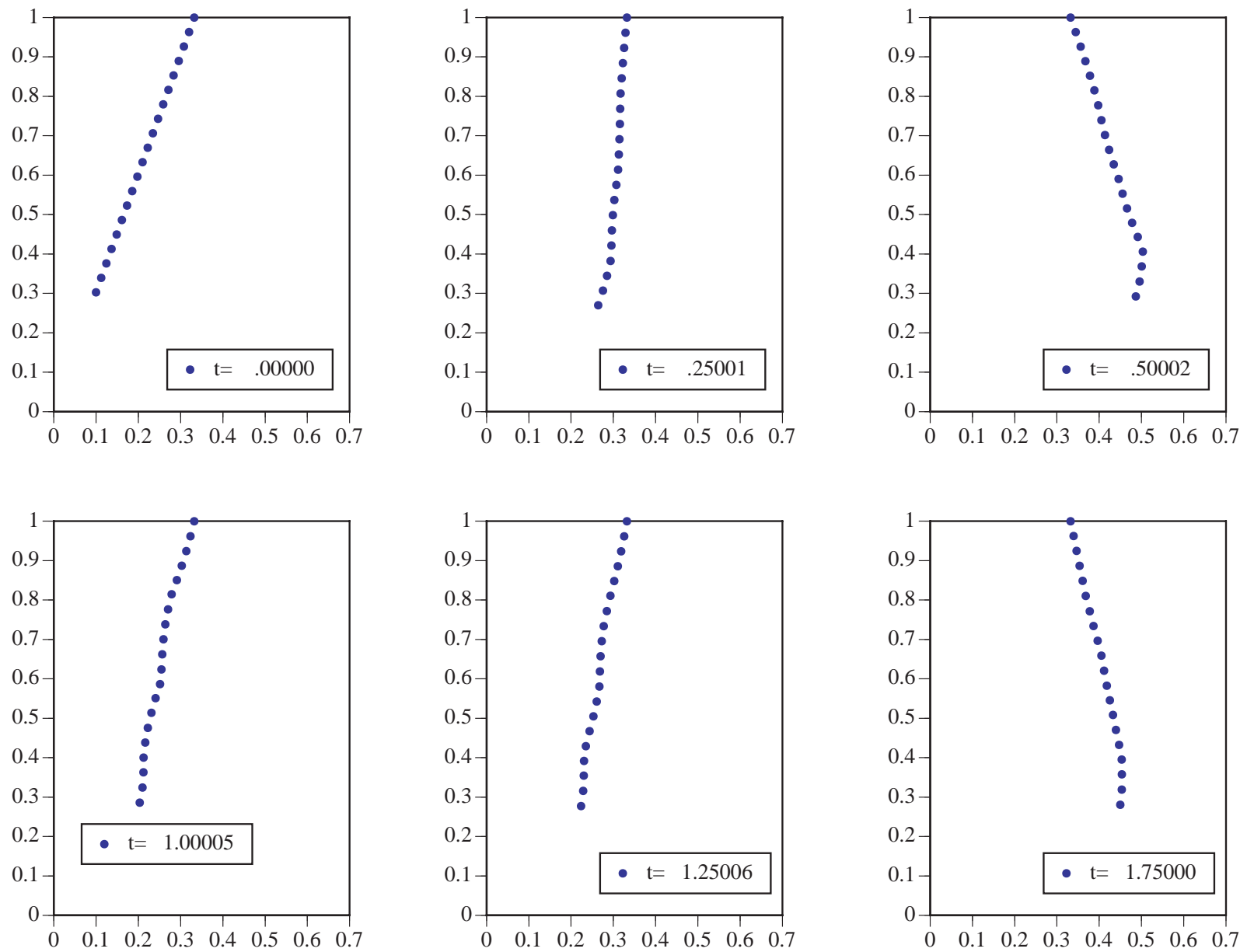


Figure 32. Pendulum Position at Various Times - MPM Simulation (20 mp)

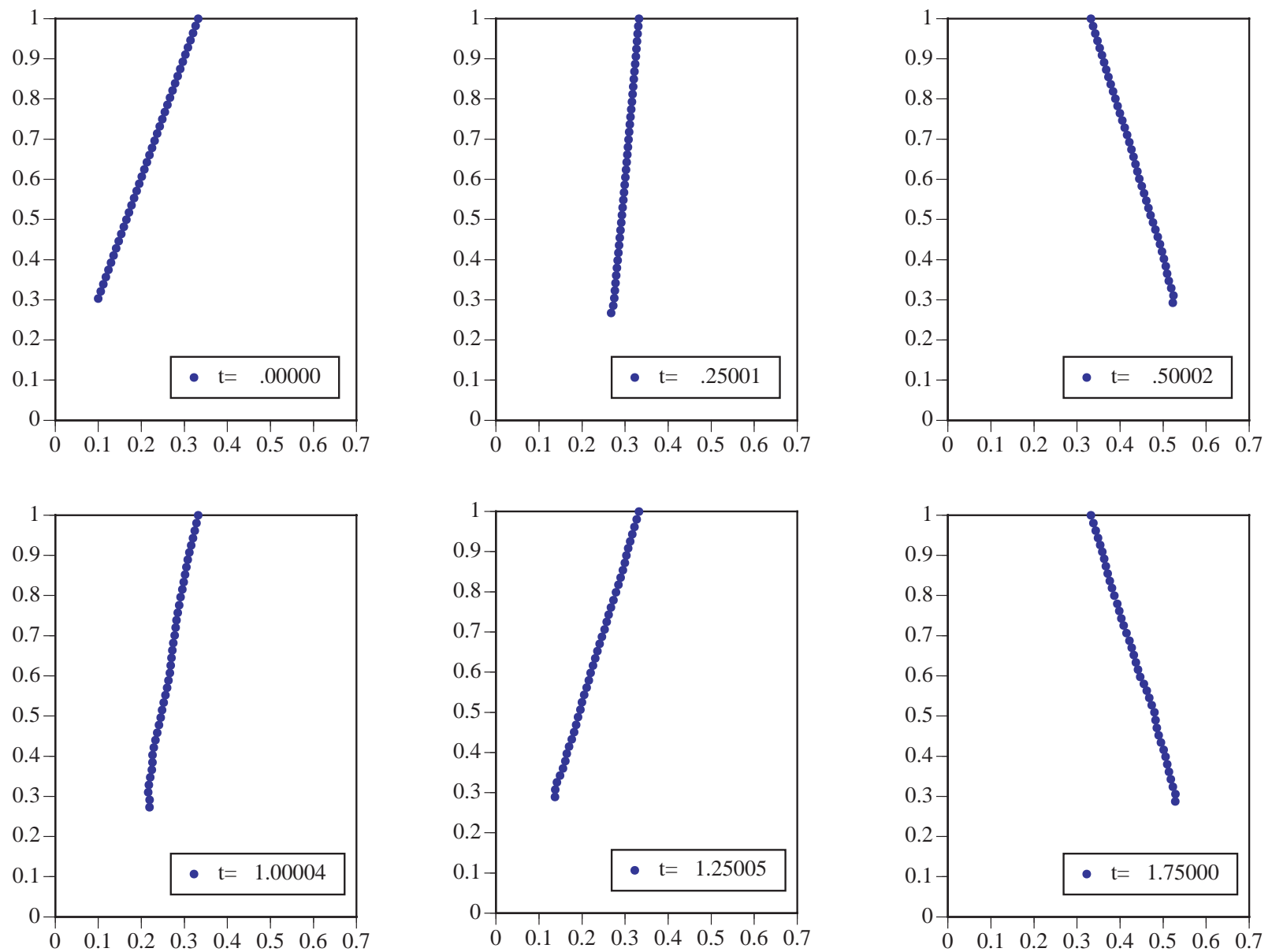


Figure 33. Pendulum Position at Various Times - MPM Simulation (40 mp)

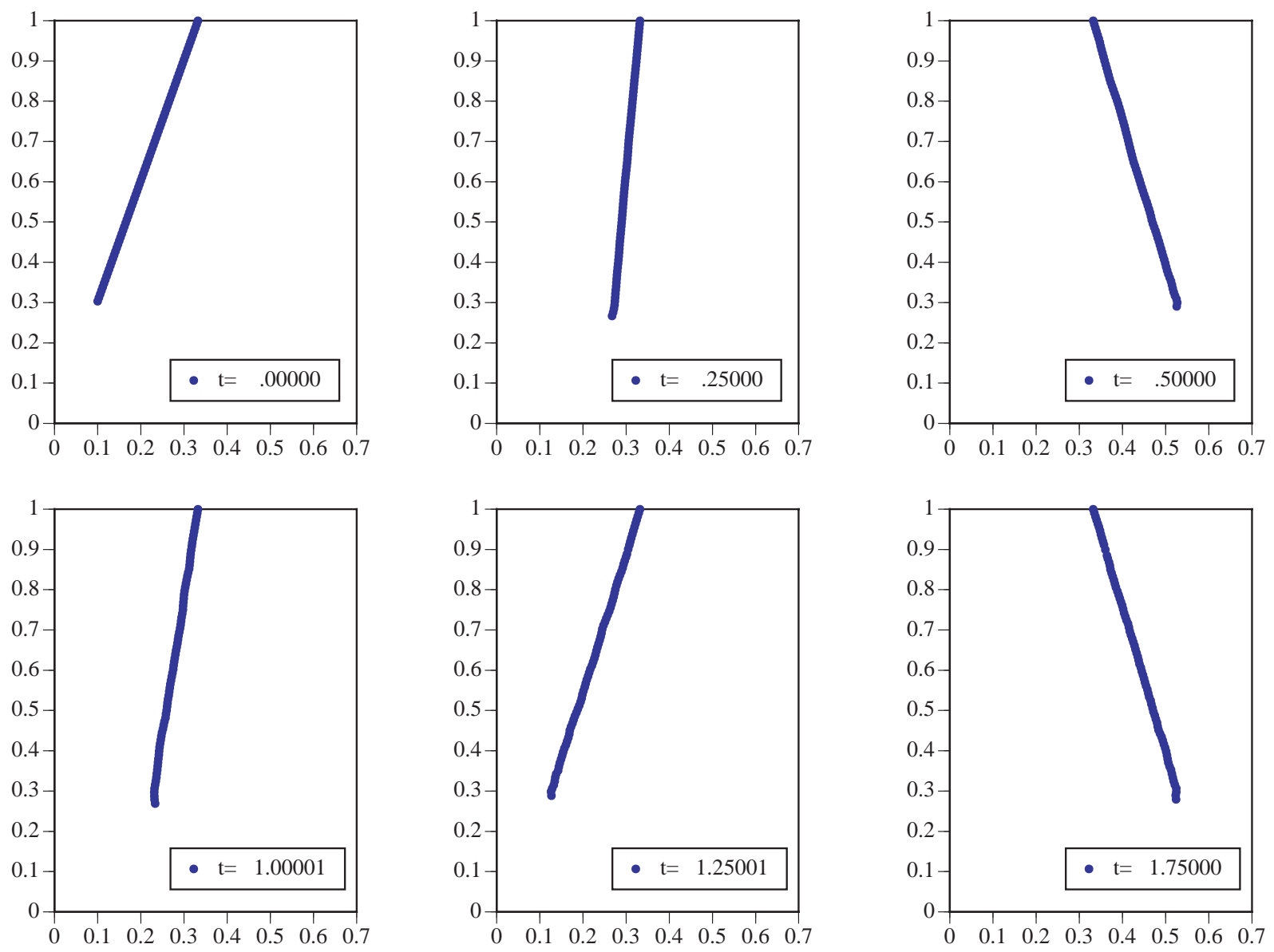


Figure 34. Pendulum Position at Various Times - MPM Simulation (80 mp)

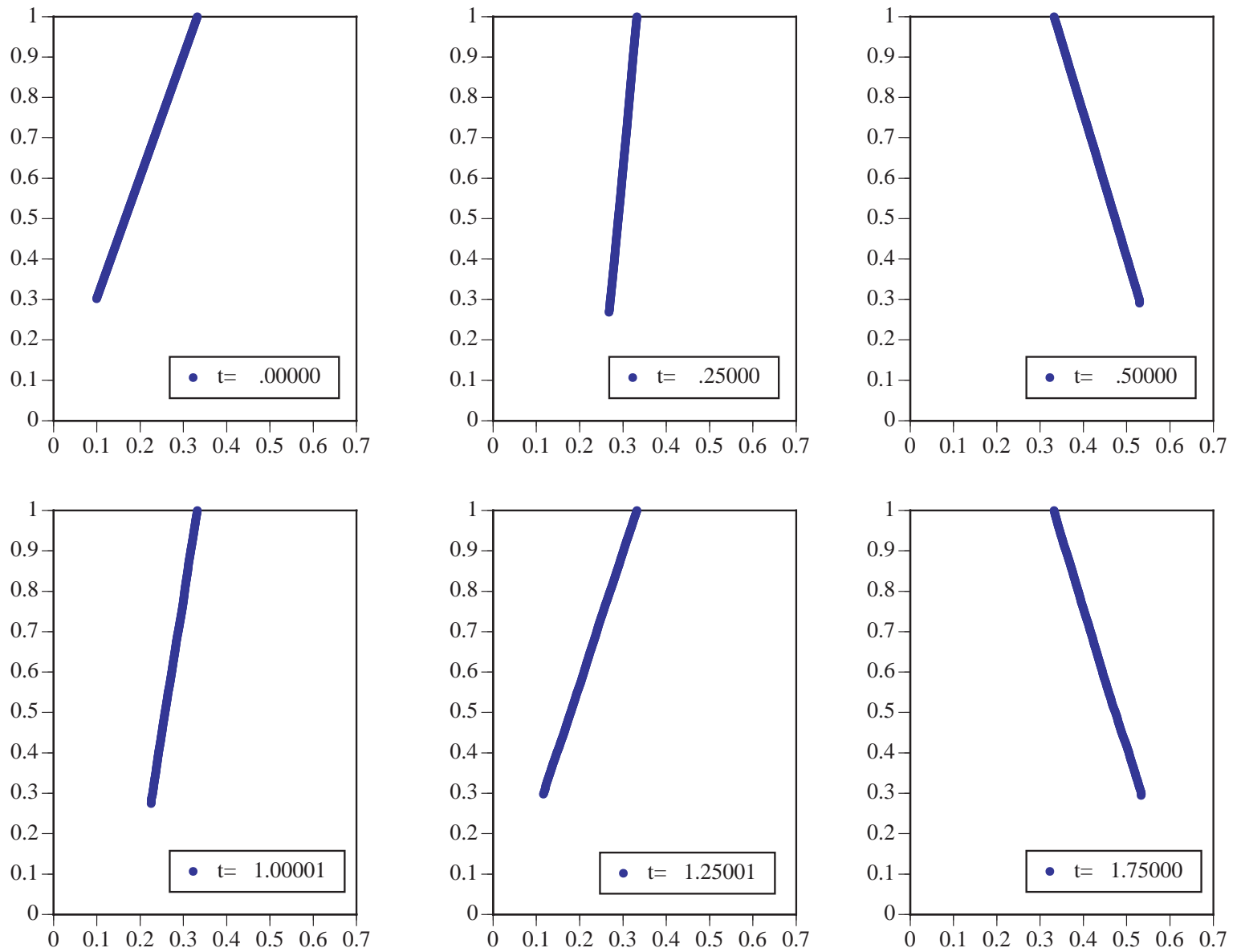


Figure 35. Pendulum Position at Various Times - MPM Simulation (320 mp)

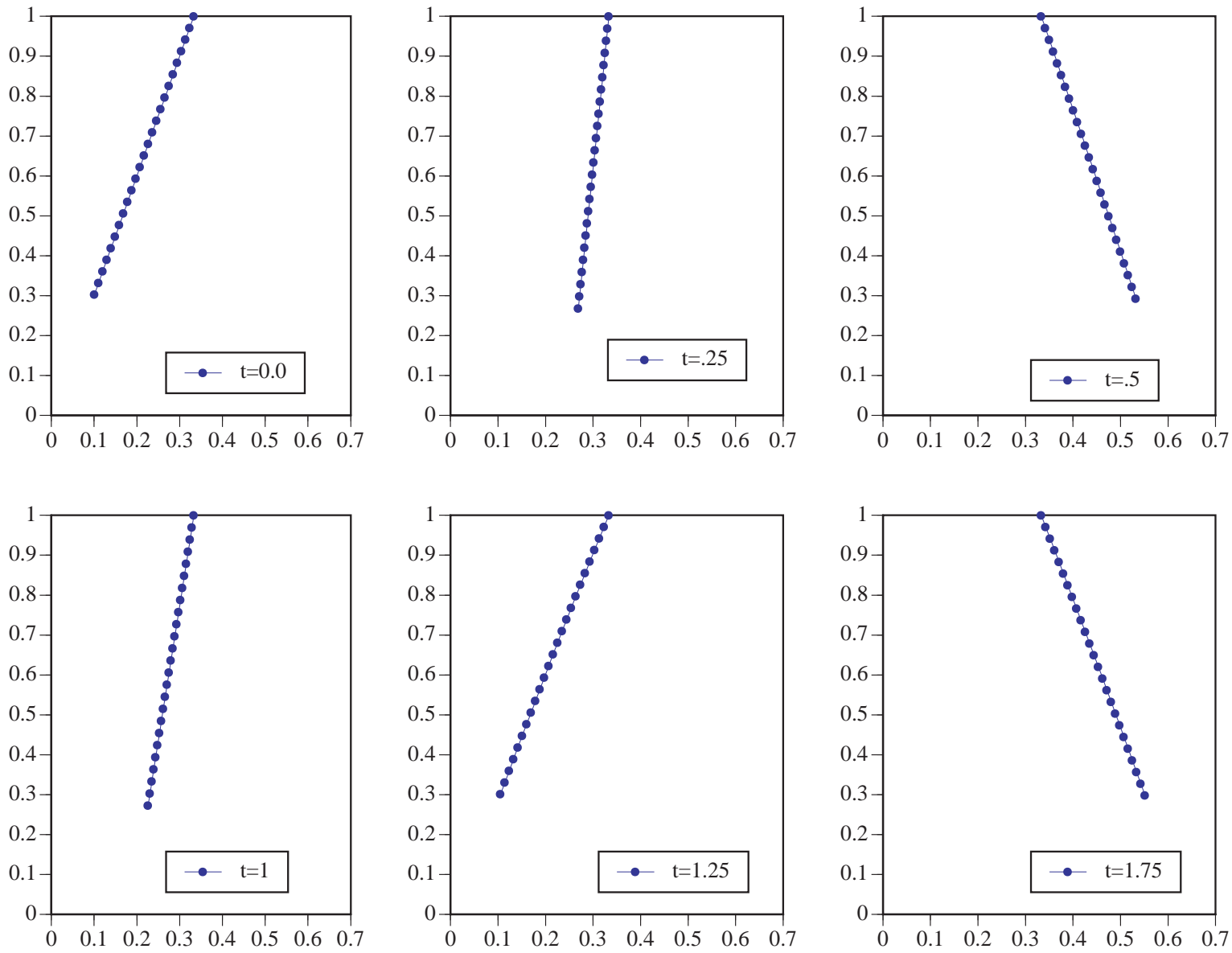


Figure 36. Pendulum Position at Various Times - FEM Simulation

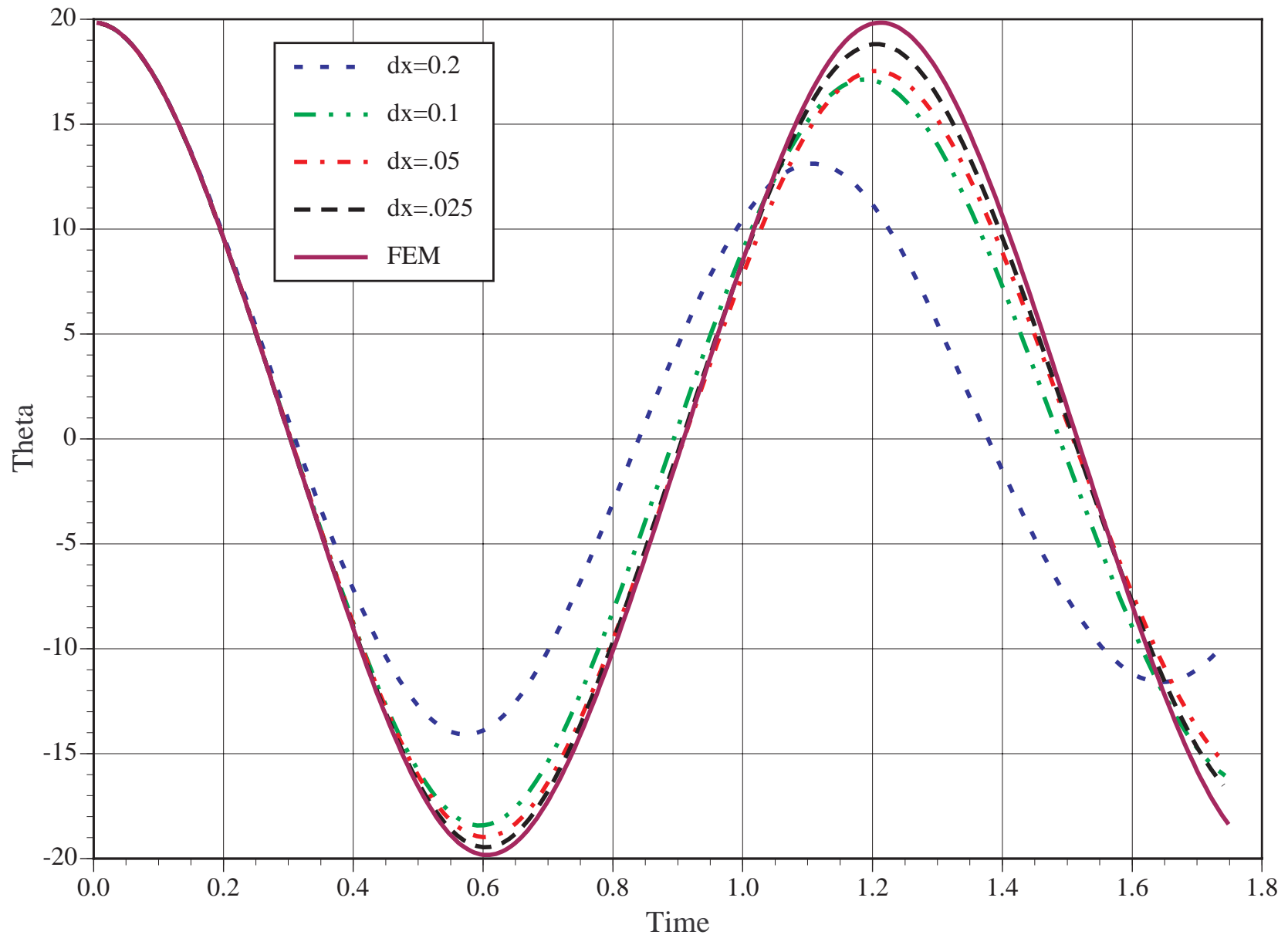


Figure 37. Comparison of Angle Theta For Swinging Pendulum Simulations

5.7.3 MPM Pendulum Simulation Results - Without Wrinkle Algorithm

If the wrinkle algorithm is not used as described in Section 5.1, the results are similar, but in some cases appear slightly nonphysical. This is why the convergence study is presented using the wrinkle algorithm. The position of the material points at various times for a simulation without the wrinkle algorithm is illustrated in Fig. 38, page 66.

Without the wrinkle algorithm the alignment of material points is not as good (Fig. 38). One possible explanation of why this misalignment occurs lies in the fact that as the pendulum swings it also oscillates due to the spring-like effect of the string. The oscillations “up” may cause some material points to go into compression. The material points cannot support compression in this configuration so a phenomenon similar to buckling may occur which tends to misalign the material points.

5.8 Plane Stress

The plane stress formulation is similar to uniaxial stress, except that other conditions on the strains are enforced. In this case, the strain in the 3' direction is set so that the stress in this direction is zero. Also, a plane strain assumption in the 2' direction dictates that the strain in this direction be zero, but the stress is nonzero. Figure 39 illustrates a small material element in a state of plane stress. These

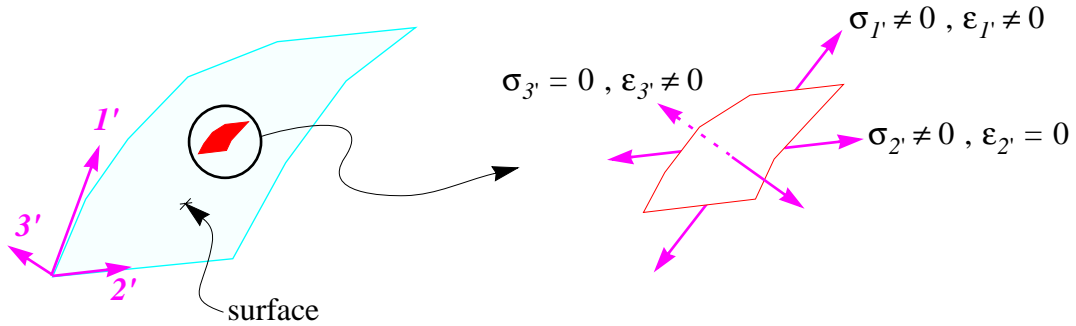


Figure 39. Plane Stress Assumptions

plane stress assumptions are

$$\epsilon_{3'} = -\nu \epsilon_{1'} / (1 - \nu) \quad \text{and} \quad \epsilon_{2'} = 0. \quad (5.12)$$

The strains in the primed (local) coordinate system are input into a matrix multiplication routine to determine the stresses as follows

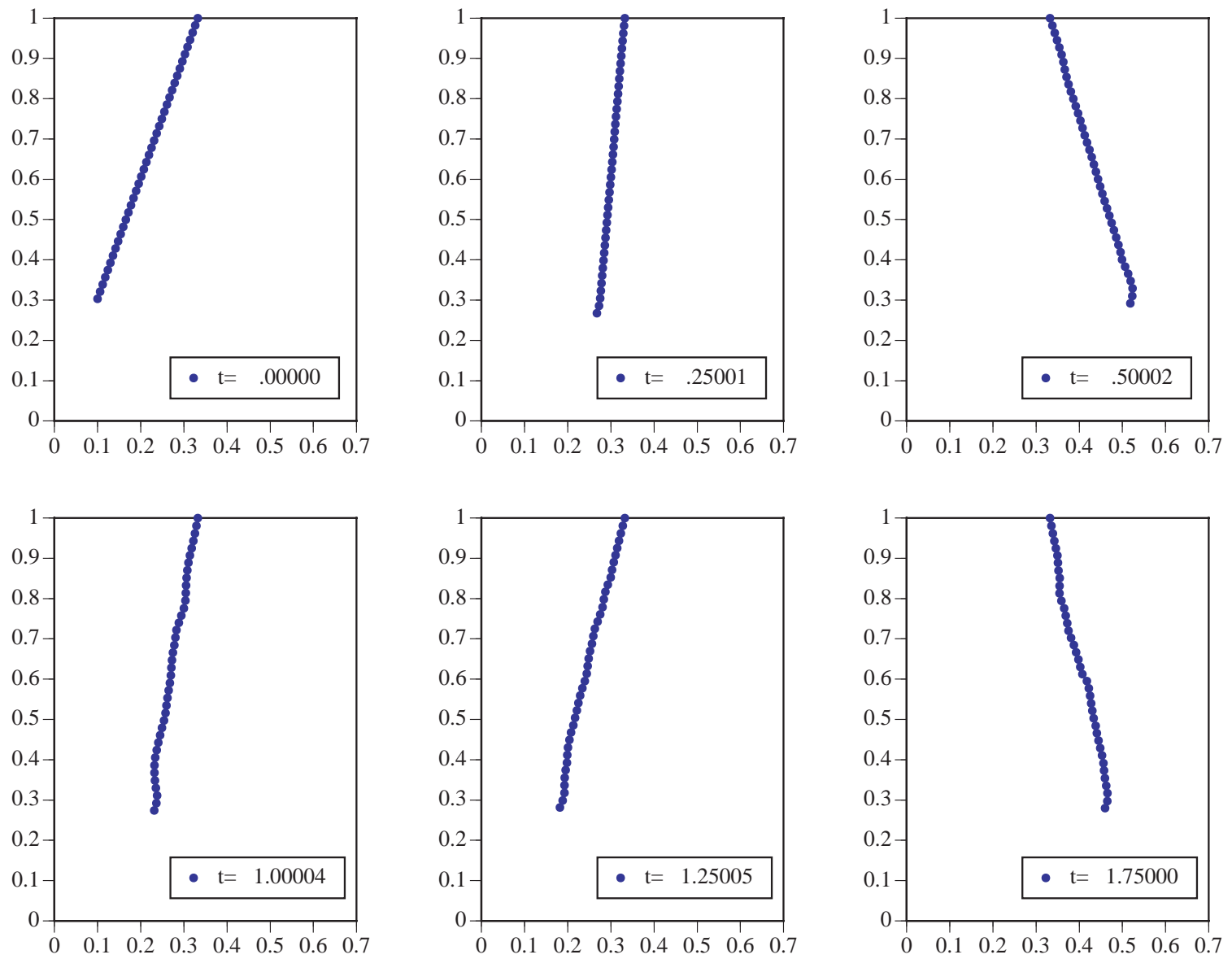


Figure 38. Pendulum Simulation (40 mp, $\Delta=0.1$) Without Wrinkle Algorithm

$$\begin{bmatrix} \sigma_{I'} \\ \sigma_{3'} \\ \sigma_{2'} \\ \sigma_{I'-3'} \end{bmatrix} = [E] \begin{bmatrix} \varepsilon_{I'} \\ \varepsilon_{3'} \\ \varepsilon_{2'} \\ \varepsilon_{I'-3'} \end{bmatrix} = [E] \begin{bmatrix} \varepsilon_{I'} \\ -\nu \varepsilon_{I'}/(1-\nu) \\ 0 \\ 0 \end{bmatrix} \quad (5.13)$$

Expanding equation 5.13 gives

$$\begin{bmatrix} \sigma_{I'} \\ \sigma_{3'} \\ \sigma_{2'} \\ \sigma_{I'-3'} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 \\ \nu & (1-\nu) & \nu & 0 \\ \nu & \nu & (1-\nu) & 0 \\ 0 & 0 & 0 & (1-2\nu) \end{bmatrix} \begin{bmatrix} \varepsilon_{I'} \\ -\nu \varepsilon_{I'}/(1-\nu) \\ 0 \\ 0 \end{bmatrix} \quad (5.14)$$

which simplifies to

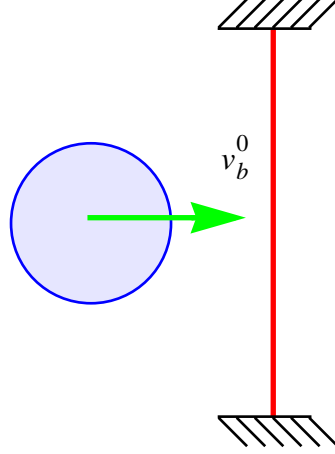
$$\begin{bmatrix} \sigma_{I'} \\ \sigma_{3'} \\ \sigma_{2'} \\ \sigma_{I'-3'} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} \varepsilon_{I'}(1-\nu) - \nu^2 \varepsilon_{I'}/(1-\nu) \\ \nu \varepsilon_{I'} - \nu \varepsilon_{I'} \\ \nu \varepsilon_{I'} - \nu^2 \varepsilon_{I'}/(1-\nu) \\ 0 \end{bmatrix} = \begin{bmatrix} E \varepsilon_{I'}(1-\nu^2) \\ 0 \\ E \varepsilon_{I'}\nu(1-\nu^2) \\ 0 \end{bmatrix}. \quad (5.15)$$

In this simplified formulation, there is no shear in the $I'-3'$ or $I'-2'$ directions. In general this is not true for the $I'-2'$ plane.

5.9 Ball and Net Simulation

A hypothetical problem of a ball impacting a stationary net was used to test the membrane plane stress formulation. The problem set-up is shown in Fig. 40. The constitutive model for the ball is plane strain and that of the net is membrane plane stress.¹ The material properties used are listed in Table 2.

1. Since plane strain is assumed, the simulation is one of a solid cylinder impacting a membrane.

**Figure 40. Ball and Net****Table 2: Parameters for the Ball and Net Simulation**

Parameter	Ball Value	Net Value
Density	1.0	0.5
Young's Modulus	1×10^4	1×10^4
Poisson's Ratio	0.3	0.0
Initial Velocity	1.0	0.0

Intuitively, one knows that the ball will impact the net; the net will displace; and the ball will release from the net to the left with a velocity smaller than its initial impact velocity. It was these common-sense features of this problem that were being tested.

With the addition of a contact-release algorithm as described in Section [A.1.1, page 154](#), the predicted positions of the ball and net seem plausible. The criteria listed in equation 9.9 of the appendix were used to control the contact and release of the ball.

The material point positions are shown in Fig. 41. At about $t=2.8$ the ball releases from the net and moves at constant velocity to the left. This is more evident in Fig. 42 where the histories of the center of mass velocity and position of the ball and net are plotted. Here it is seen that the ball's velocity is constant at about -0.7 when $t \approx 2.8$, and the slope of the ball's position curve is constant indicating constant velocity. The net continues to oscillate after the ball releases.

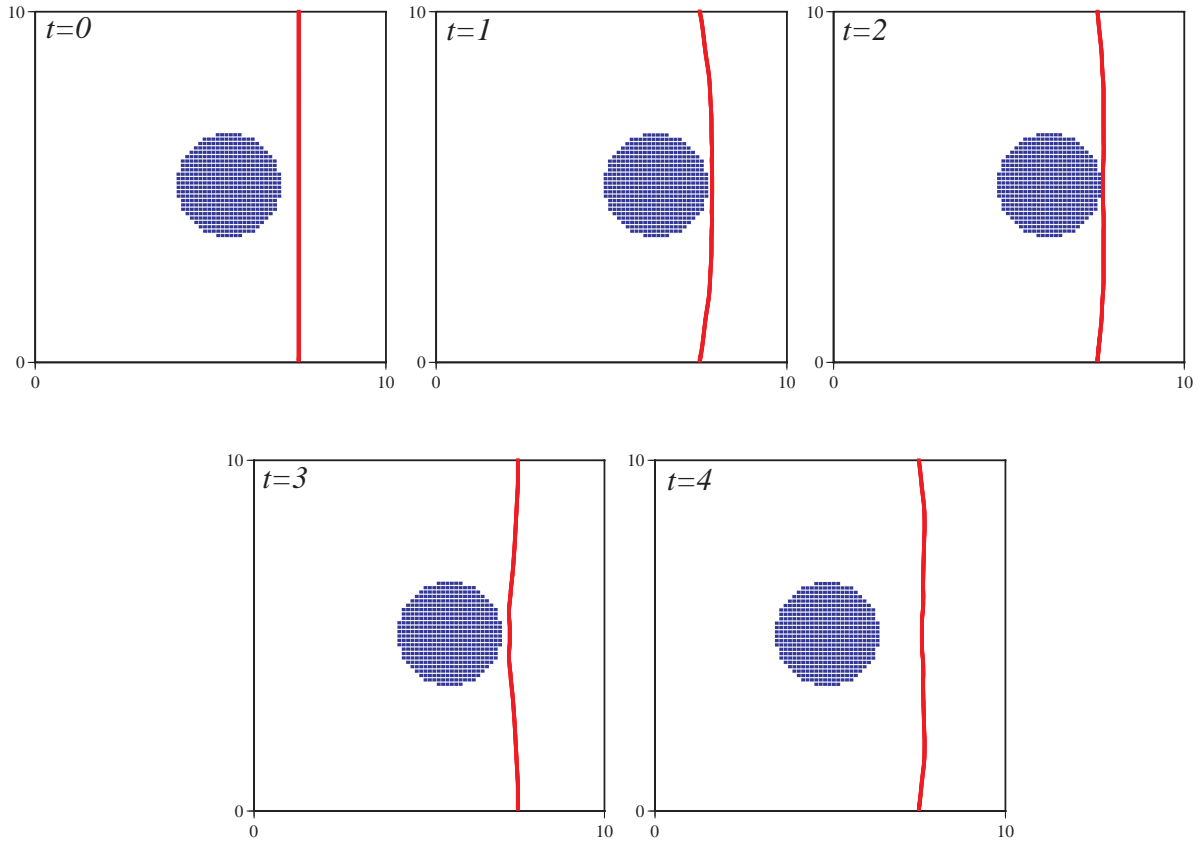


Figure 41. Ball and Net Material Point Positions at Various Times

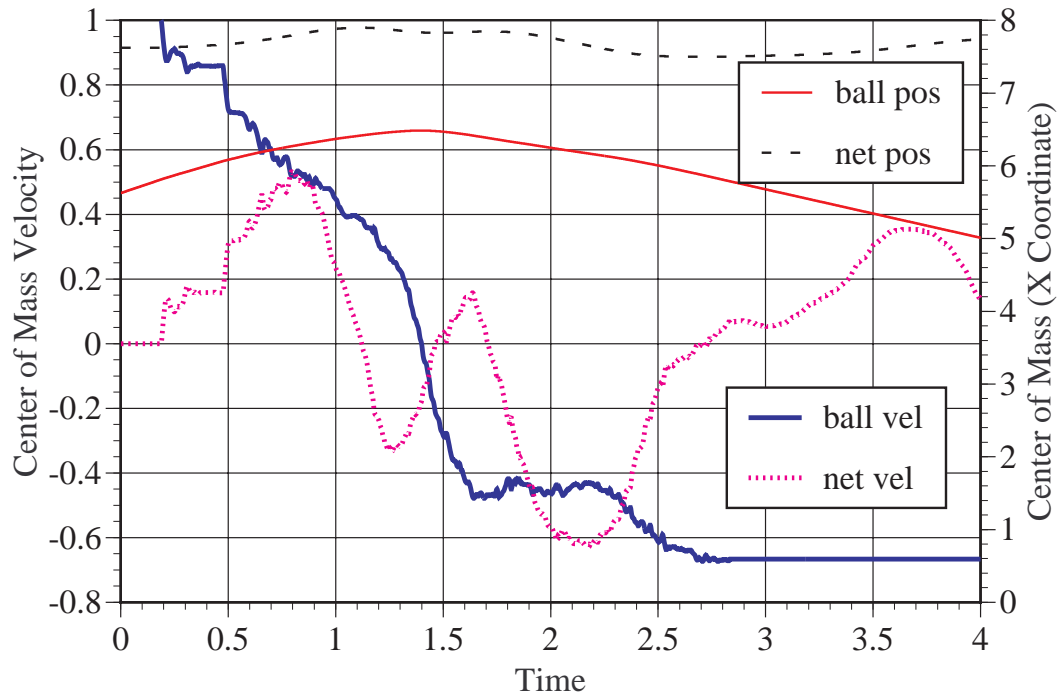


Figure 42. Center of Mass Velocities and Positions

5.10 Chapter 5 Summary

Simulations of membranes have been performed with the material point method. A relatively simple modification to the material point constitutive equation allows springs, strings, and membranes to be simulated. The spring-mass-with-slack system and ball and net simulations give qualitatively correct results. The results of the other simulations compare favorably to theory and to other simulations. This chapter shows that the MPM is a viable method for simulation of membranes. The finite element method may be more efficient for simulating some of the problems discussed here. However, the strength of this approach is treating more complicated fluid-structure interaction efficiently. The next chapter combines the fluid and membrane formulations and discusses fluid-structure interaction.

CHAPTER 6. FLUID-STRUCTURE INTERACTION WITH THE MPM

It is proposed that fluid-structure interaction problems be simulated with the MPM. The idea is straight forward in that the coupled problem is set up as any other type of MPM simulation. The difference now being that some material points are designated membrane material points and others are designated fluid material points.

The effect of the fluid on the structure and vice-versa will be determined on the grid when the equations of motion are solved at each grid node. The coupling of the fluid and solid is indirect in the sense that the pressure from a fluid material point is not directly applied to the neighboring structure material points. Instead, the forces from fluid and solid material points are calculated together at grid nodes where the divergence of the material point stress is summed. Figure 43 illustrates the idea of summation of the grid forces, f_g , from the fluid and membrane stresses, σ_f and σ_m , and the respective material point volumes, $V_{f,p}$ and $V_{m,p}$.

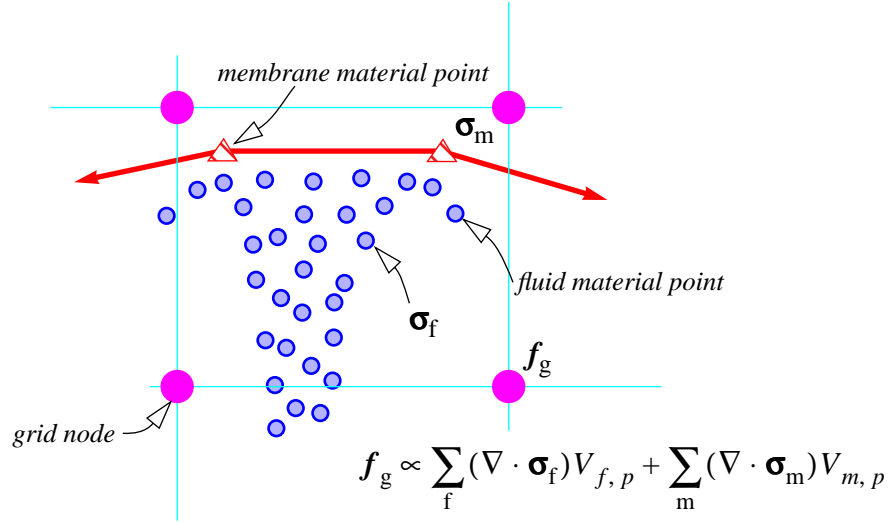


Figure 43. Fluid-Structure Coupling

The net effect of the force summation is that the structure forces cause accelerations of neighboring fluid and structure material points, and pressure and fluid shear stresses cause accelerations of neighboring structure and fluid material points. The accelerations are imposed on the material points as velocities in the calculations.

Since Lagrangian material points are used for both the fluid and the structure and since the two are indirectly coupled, the time-consuming calculations involved in defining the interface and applying

the correct boundary conditions can be avoided. Recall in Section 2.2 that one author reported calculating the intersection of the Lagrangian elements with the Euler fluid elements took the majority of the cpu time in his fluid-structure interaction calculation. Also, mixed cells are handled naturally as the material points carry material properties with them.

6.1 Piston-Container Problem

The piston-container problem is a relatively simple one-dimensional problem. The objective of running this simulation was to test the fluid-structure interaction algorithm on a simple scale using a test problem with a known solution.

6.1.1 Problem Description

An illustration of the piston-container fluid-structure problem is shown in Fig. 44. A massless spring of constant K and length L is attached to a piston. The piston can move without friction to compress or expand a compressible fluid of density ρ and bulk modulus β . A similar fluid-structure problem is presented by Olson (1983). The objective is to compare the theoretical frequency of vibration with that from an MPM simulation.

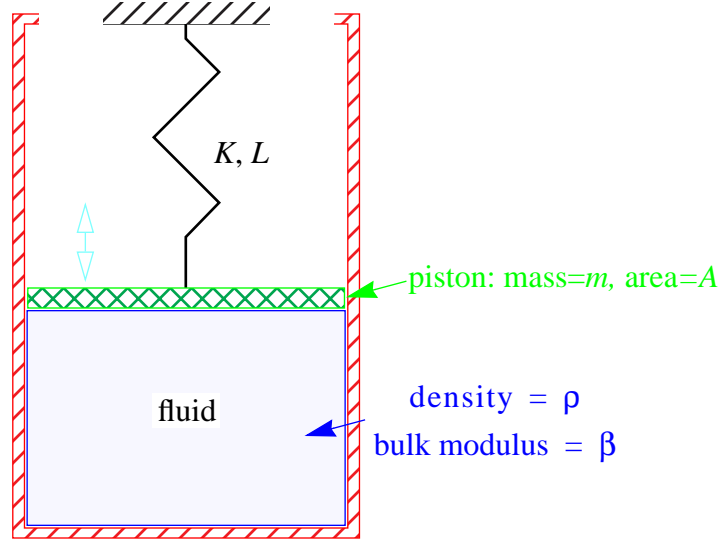


Figure 44. Piston-Container Problem

The fluid is assumed to be compressible, inviscid, and adiabatic. The continuity, energy, and constitutive relations combine to give the relationship between pressure and displacement as follows:

$$p = -\beta(\nabla \cdot \mathbf{U}) \quad (6.1)$$

where U is displacement and β is the isentropic bulk modulus. The analytical expression for the natural frequency, ω , of vibration of the mass satisfies the equation

$$\omega - \sqrt{[K + \rho \omega c A \cot(\omega L/c)]/m} = 0 \quad (6.2)$$

where c is the wave speed $\sqrt{\beta/\rho}$, and A is the piston cross-sectional area.

6.1.2 Piston-Container Simulation Results

The parameters used in the MPM simulation are as follows: $L=20$, $A=1.0$, $K=100$, $\rho=0.0001$, $\beta = 1.58 \times 10^6$, and m variable. Compared with Olson's paper, the area here is reduced by a factor of 5, and β is increased by a factor of 5. The density in the simulation is $1e-4$ compared with $9.35e-5$ used in Olson's problem. The problem is set up with the MPM simulation as illustrated in Fig. 45. A line of code was added to the equation of state subroutine in the MPM program to determine the pressure as indicated in equation 6.1. Also, the heavy material point in the spring is "hardwired" to have a mass of 100 times that of other spring material points, which is an approximation of the massless spring.

To determine the frequency of vibration, the heavy material point representing the piston is given a small initial velocity, and the position of this material point is monitored.

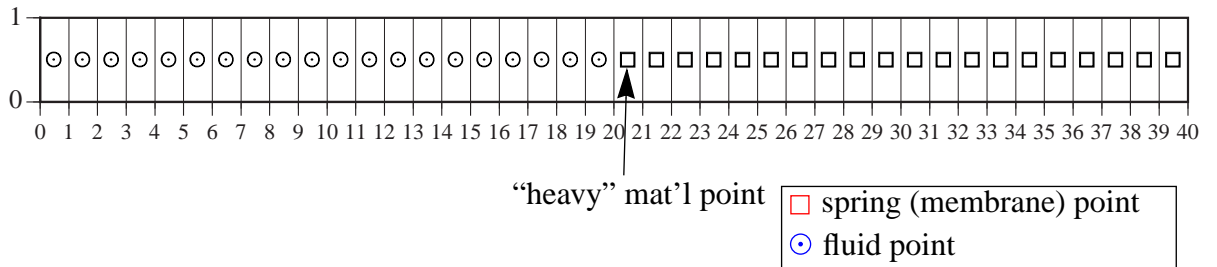


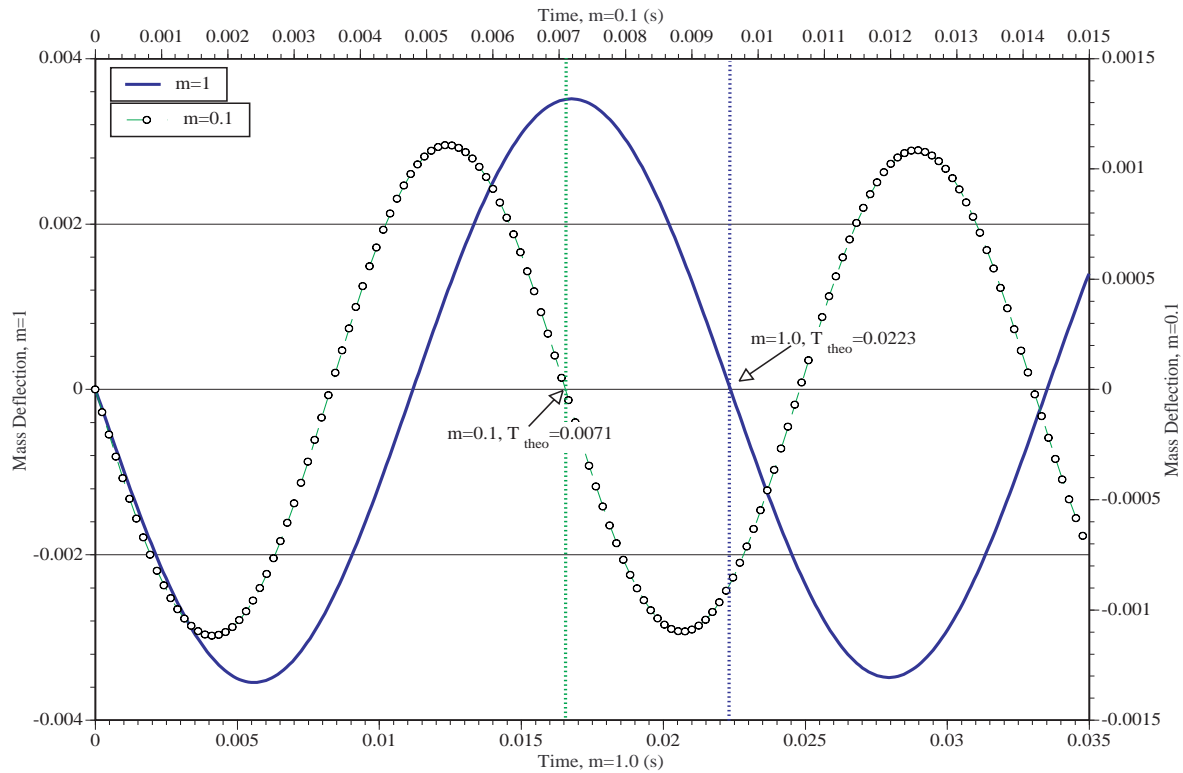
Figure 45. Piston-Container MPM Simulation Set-up

The first solution of equation 6.2 gives the fundamental mode of vibration. Table 3 lists the theoretical natural frequency and period for a given mass along with the period observed in the MPM simulation. The observed periods are very close to the theoretical ones. Figure 46 shows the time history of displacement of the mass. The vertical dashed lines in the plot represent the theoretical periods of vibration.

Table 3: Piston-Container Periods of Vibration

mass, m	Theoretical ω (rad/s)	Theoretical period (s)	MPM Simulation period (s)
0.1	886.2	0.00709	0.0071
1.0	281.2	0.02234	0.022

The period of vibration with a mass of 1.0 in a vacuum (no fluid) is 0.628 s, so the effect of the fluid in the case of $m=1$ is to increase the frequency of vibration by a factor of nearly 30.

**Figure 46. Mass (Piston) Deflection in the Piston-Container Simulations**

6.2 Membrane Expansion

The dog-bone-to-cylinder membrane expansion simulation is an arbitrary problem (not seen in the literature) that tests several aspects of fluid-structure interaction. The initial conditions for the problem are shown in Fig. 47, and the problem parameters are listed in Table 4. A computational cell size of 0.1 is shown in the figure; a total of four different cell sizes were used.

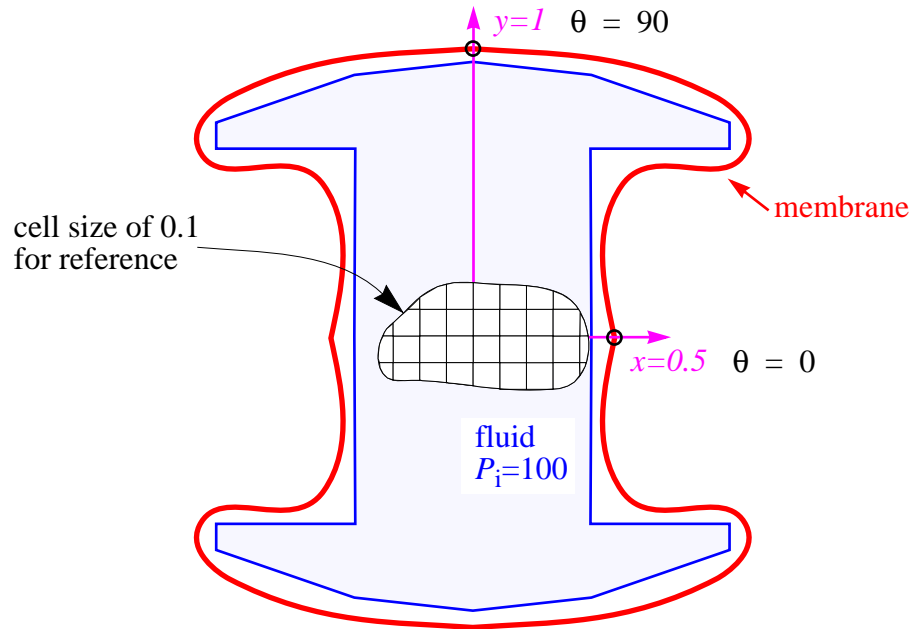


Figure 47. Initial Conditions for the Dog-bone Membrane Expansion Simulation

Table 4: Fluid and Membrane Properties

Fluid Property	Value	Membrane Property	Value
density	1.0	density	0.5
viscosity, μ	0.1	Young's modulus	1×10^6
initial specific internal energy, i	250	Poisson's ratio, ν	0.3
ratio of specific heats, γ	1.4	wrinkle algorithm	on
artificial viscosity	off		

The problem consists of a gas under internal pressure which expands filling a cavity that is initially dog-bone shaped. A membrane forms the cavity and confines the gas. At early times, the membrane oscillates due to unbalanced forces. Since the fluid is given a nonzero viscosity coefficient, the membrane oscillations are damped until a steady state condition is reached.

There are several areas where this problem can be compared to theory. Some of these are as follows:

- the gas should not escape from the membrane,
- the equilibrium shape of the membrane should be circular,
- the stress in the membrane should be consistent with the internal pressure.

Table 5 lists some of the results of the four simulations that were performed. The simulations are more refined going from 1 to 4 as indicated by the cell or mesh size in the second column. Square elements are used. The last two columns show the final pressure and radius of the membrane at 0 and 90 degrees (see Fig. 47 for 0 and 90 degree locations). The final radii of the membrane show convergence with mesh refinement, and the final pressures do also, but convergence is not as rapid as for the radii. The higher final pressures of the more refined simulations are consistent with less energy dissipation observed in the plots of the total energy.

Table 5: Membrane Expansion Simulation Parameters and Results

Simulation	Cell Size	No. Mat'l Points Fluid ^a Membrane ^b		Total Mass	Final Radius (0°/90°)	Final Internal Pressure ^c
1	0.4	64	56	0.600	1.08/1.13	33.4
2	0.2	64	100	0.584	1.335/1.334	24.9
3	0.1	64	200	0.583	1.388/1.387	27.0
4	0.05	64	400	0.584	1.394/1.396	29.0

a. Material points per cell.

b. Total number of membrane material points.

c. Calculated by averaging the final pressure at interior grid points.

Table 6 is a comparison of the theoretical hoop stress, calculated with the final radius and pressure, with the hoop stress observed in the simulations. There is nominally a ten percent difference between the theoretical value and the simulation value. One source of error arises from the fact that the

forces are calculated at grid nodes, and the radius used for the hoop stress calculation in the first column is based on material point locations.

Table 6: Comparison of Hoop Stress

Simulation	Hoop Stress ^a	Hoop Stress - Simulation	% Difference
1	369	394	6.75
2	332	372	11.95
3	375	417	11.31
4	405	447	10.49

a. Hoop stress, pr/t , is calculated using the final radius and pressure from Table 5 with thickness equal to 0.1.

Figures 48 through 55 illustrate the detailed results of the four membrane expansion simulations. For each simulation, the material point positions during the initial oscillations are shown along with the shape at steady state. Following this is a plot of the radii at zero and 90 degrees and the energy.

6.2.1 Simulation 1

Simulation 1 is the coarsest simulation with an element size of 0.4. Figure 48 shows the material point locations at increasing times. The simulation is so coarse that the folds or wrinkles in the membrane can not be pulled out to a circular shape at equilibrium. However, it can be seen that gas does not escape the membrane and that the membrane shape does move toward a circular shape although the circular shape is not totally achieved. Figure 49 shows the time history plots of radii and energy. The radii plot shows the noncircular shape at equilibrium, and the oscillations appear quite noisy. The energy plot shows that about 3.6 percent of the energy is lost due to numerical dissipation.

Some trends to look for in the upcoming plots of solutions with mesh refinement are: (i) the final shape of the membrane becomes more circular, (ii) the oscillations are smoother, and (iii) there is less energy dissipation.

6.2.2 Simulation 2

Figure 50 shows the material point locations at increasing times for the simulation where the

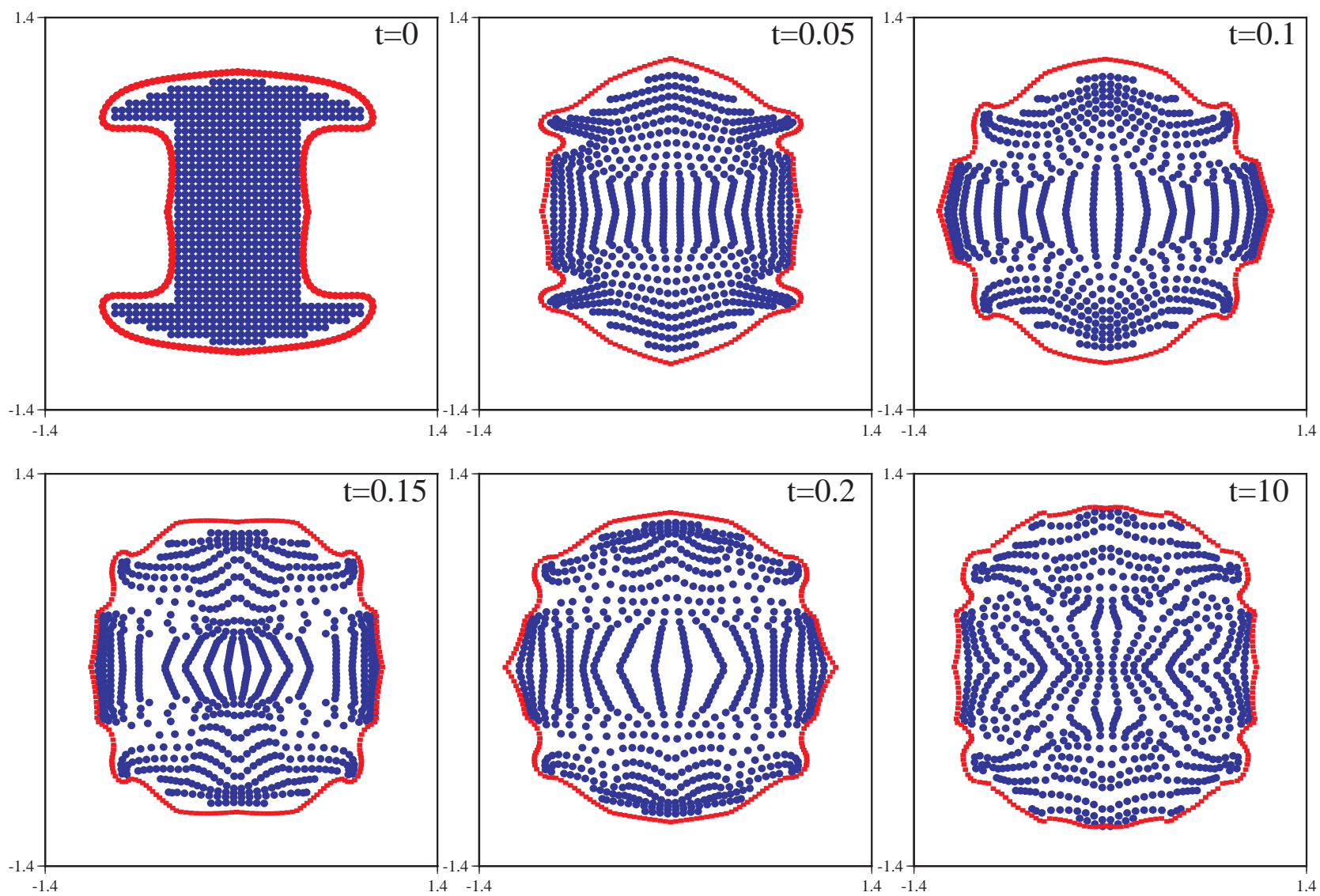


Figure 48. Material Point Position Plots for Simulation 1

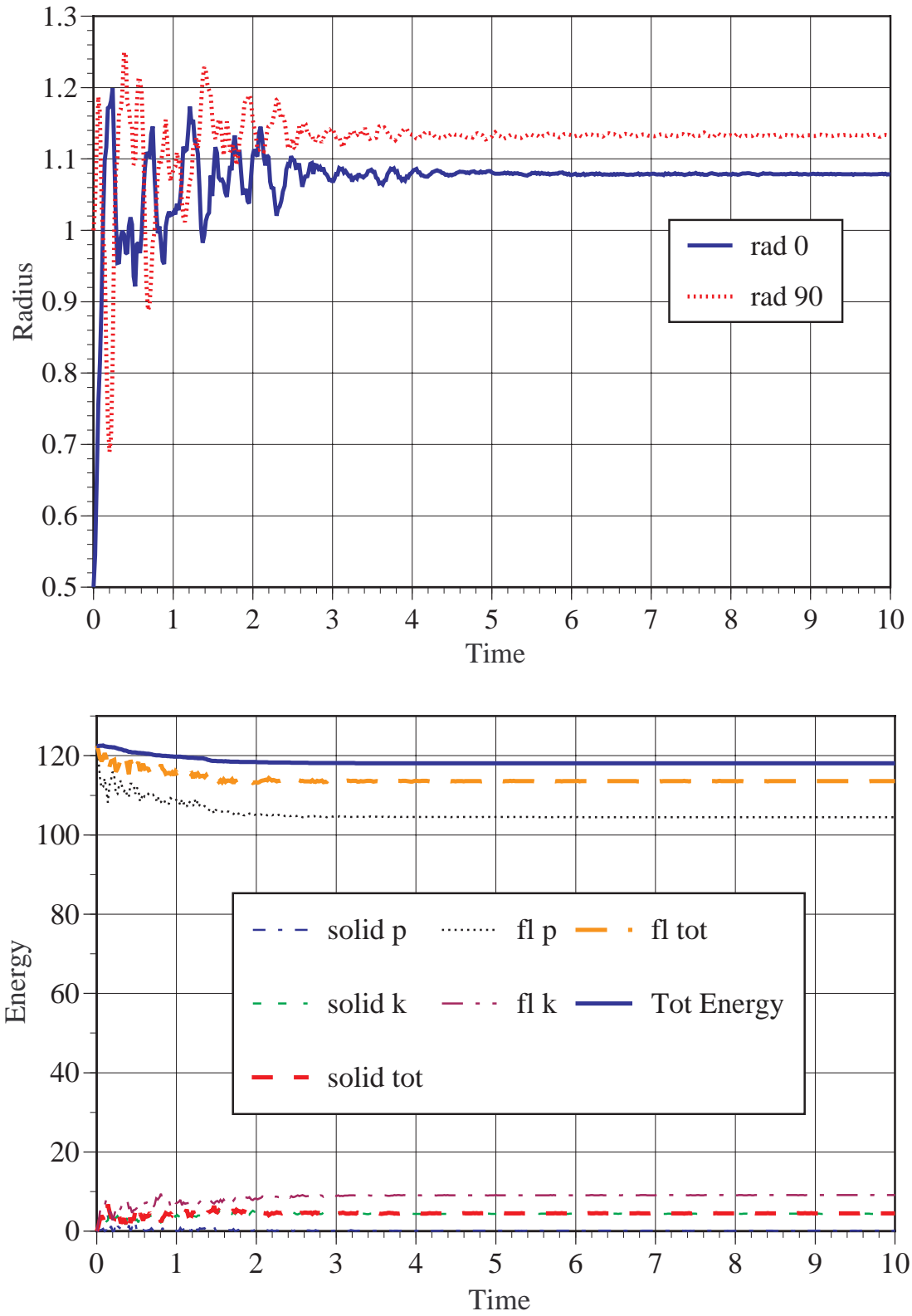


Figure 49. Radii and Energy for Membrane Expansion Simulation 1

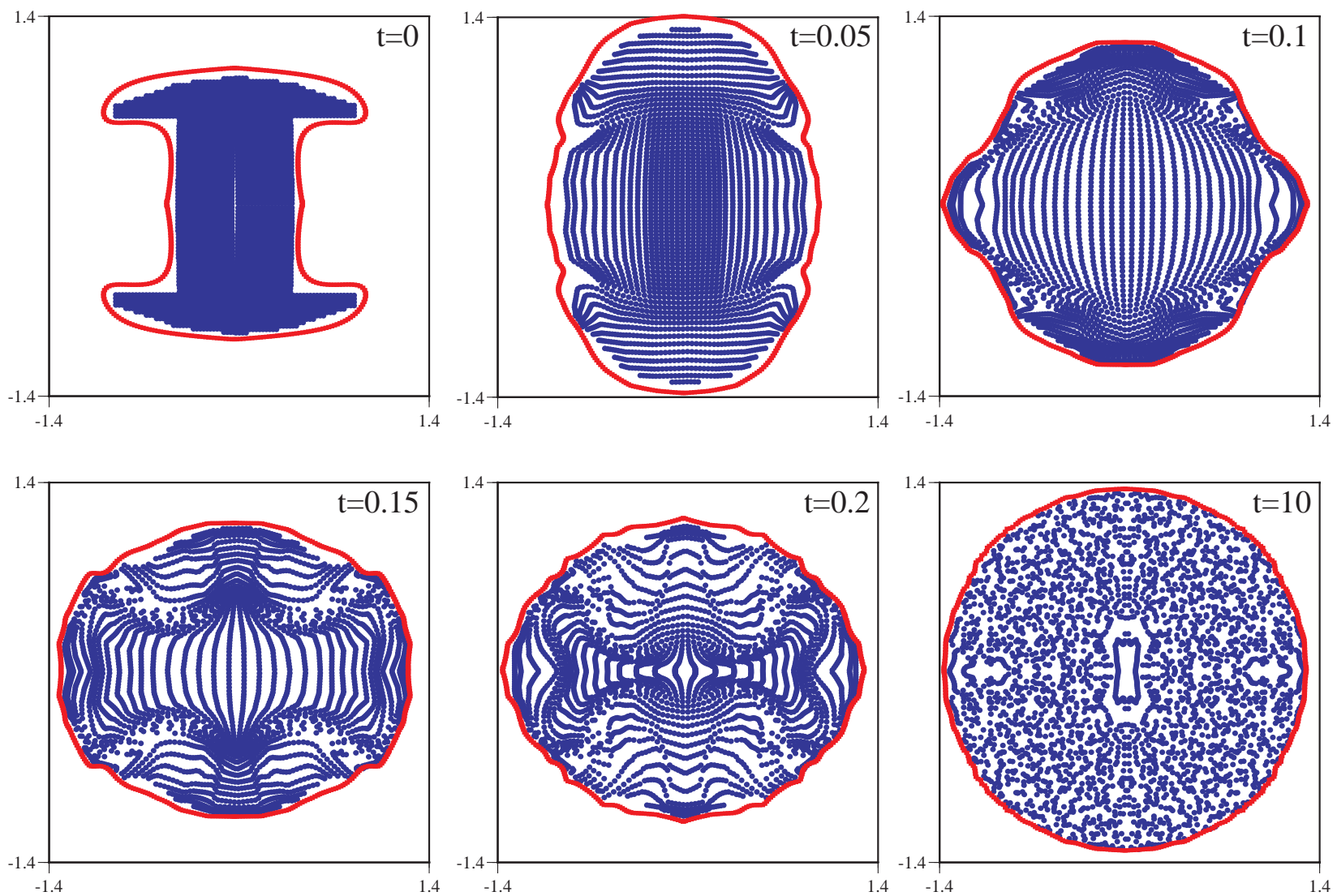


Figure 50. Material Point Position Plots for Simulation 2

mesh size is one-half of simulation 1, or $\Delta x = 0.2$. The results appear more physically intuitive than the previous one. However, one might question the bulges in the membrane at $t=0.1$ as being non-physical. In general, the membrane oscillations are reasonable, and the final shape is approximately circular. Figure 51 shows that the radii are approximately equal at zero and 90 degrees, and the energy plot shows that about 3.4 percent of the energy is lost due to dissipation.

6.2.3 Simulation 3

Figure 52 shows the material point locations at increasing times for a mesh size of 0.1. The format of this figure is slightly different from that of previous material point plots because a different post-processing technique was required due to the large number of material points. Note also that the bulges at $t=0.1$ do not appear in this simulation which suggests that they may have been due to poor resolution in the previous simulation. Also, the smoother curves in the membrane are probably attributable to higher resolution.

Figure 53 shows the radii and energy plots. The oscillations in the membrane are less noisy than the previous ones, and the change in energy is only about 1.2 percent.

6.2.4 Simulation 4

Figure 54 shows the material point locations at increasing times for the simulation with the highest resolution (mesh size 0.05). The trends in changes between this simulation and the previous one are similar to changes observed between simulations 2 and 3. The membrane lines are smoother, but the general shape is approximately the same as in Simulation 3. The same applies to the radii and energy as seen in Fig. 55. The oscillations in the radii are smoother and the change in energy is less than one percent.

The pressure and membrane stress contours are shown in Fig. 56 for quadrant I (0 to 90°). Both pressure and stress are interpolated to grid points from material points. There is no smoothing of the contours done by the plotting package; thus, some of the contours are jagged.

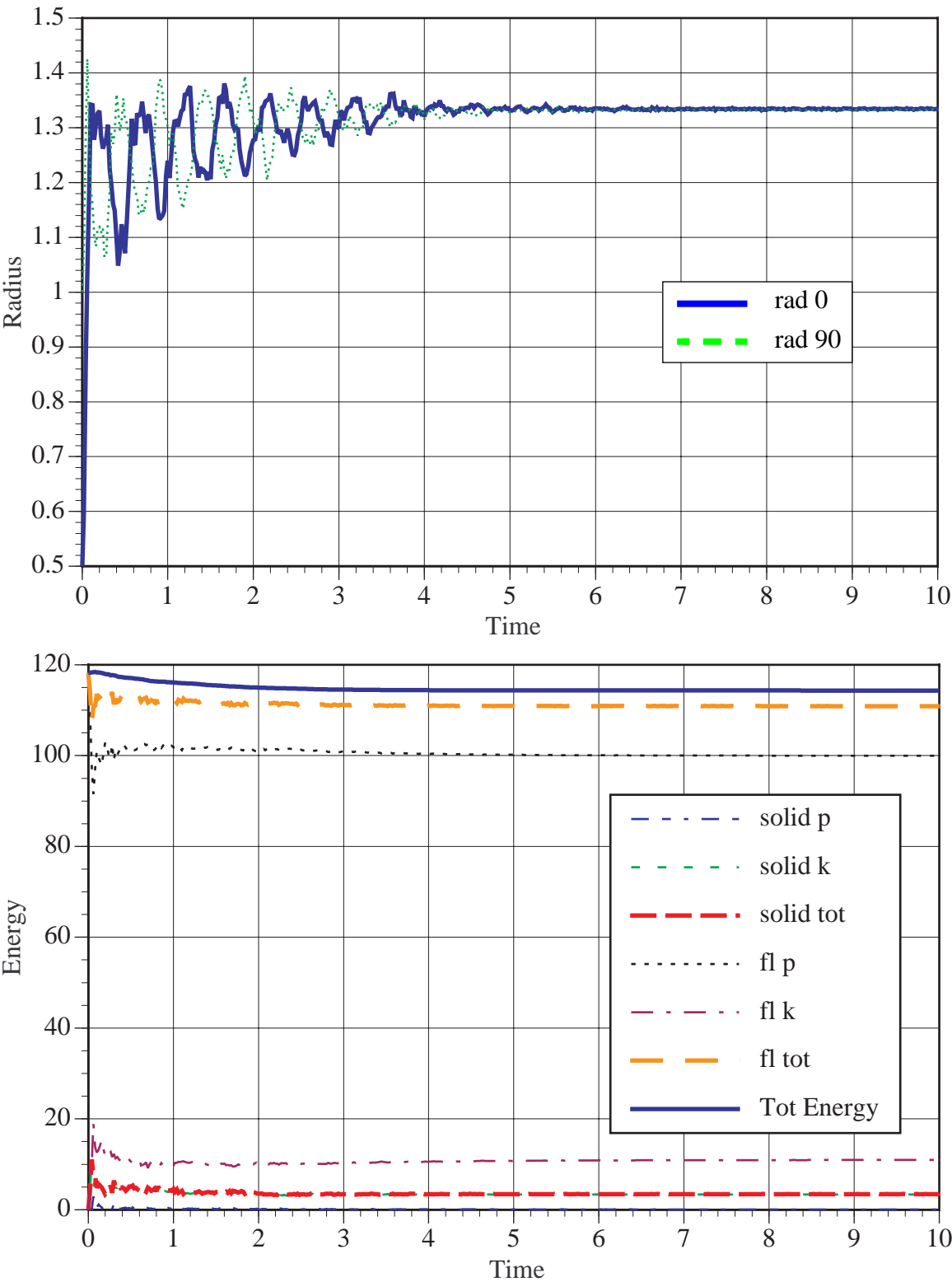


Figure 51. Radii and Energy for Membrane Expansion Simulation 2

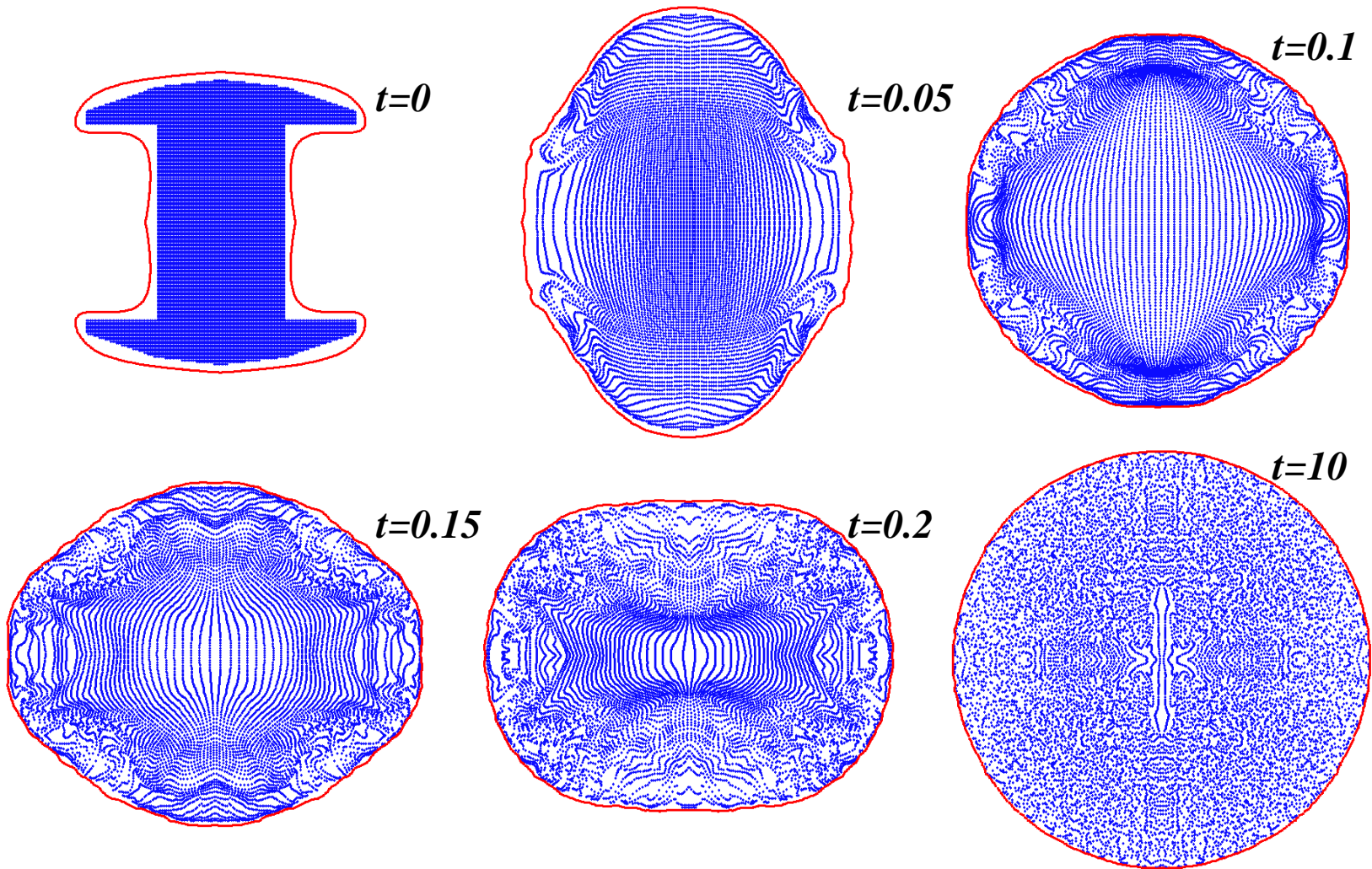


Figure 52. Material Point Position Plots for Simulation 3

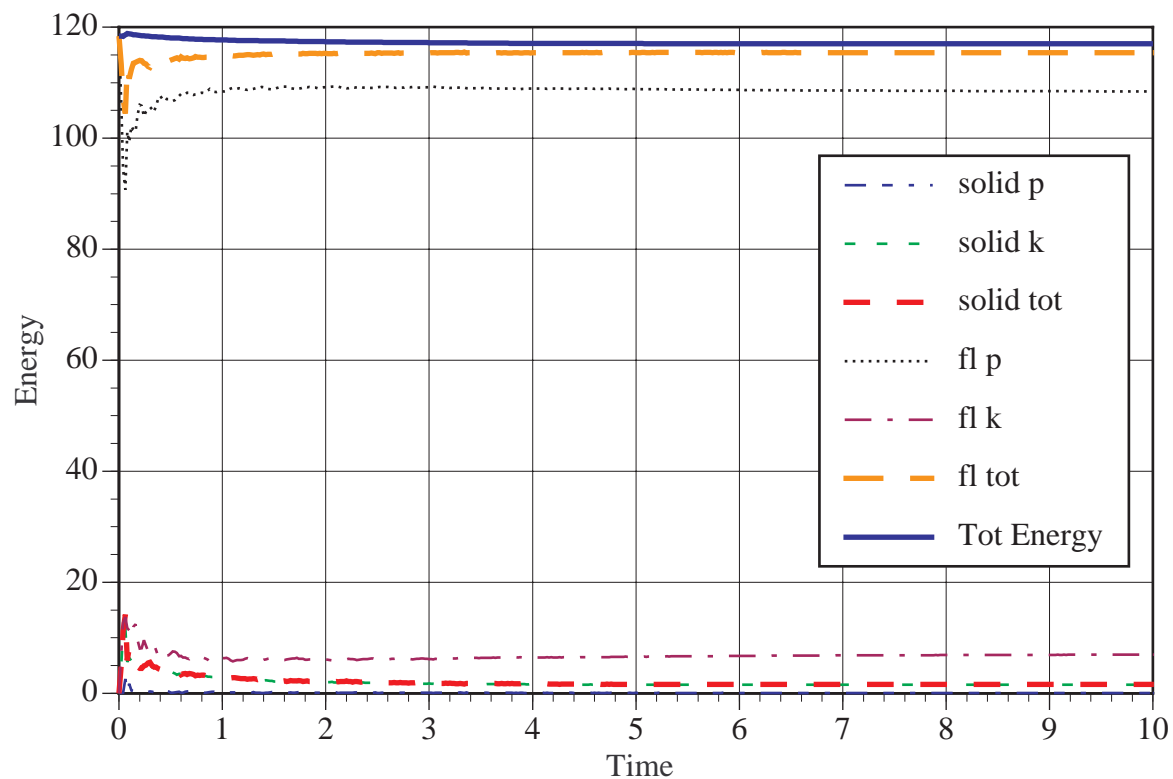
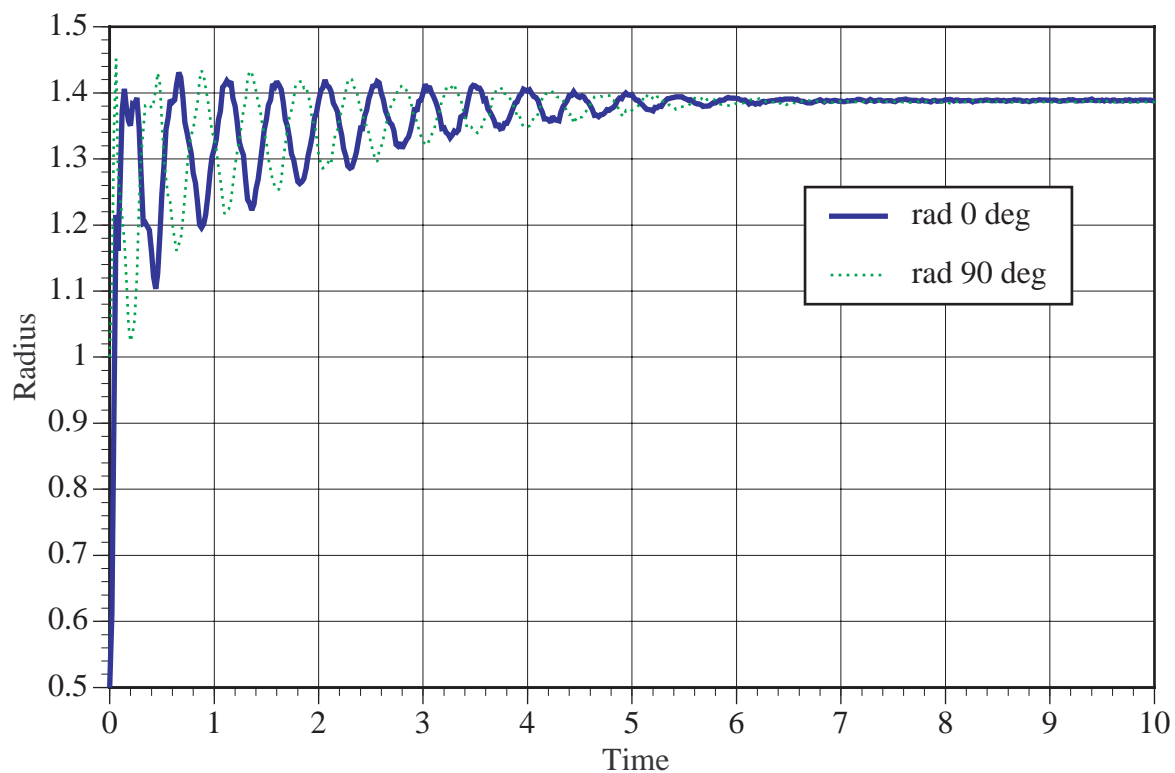


Figure 53. Radii and Energy for Membrane Expansion Simulation 3

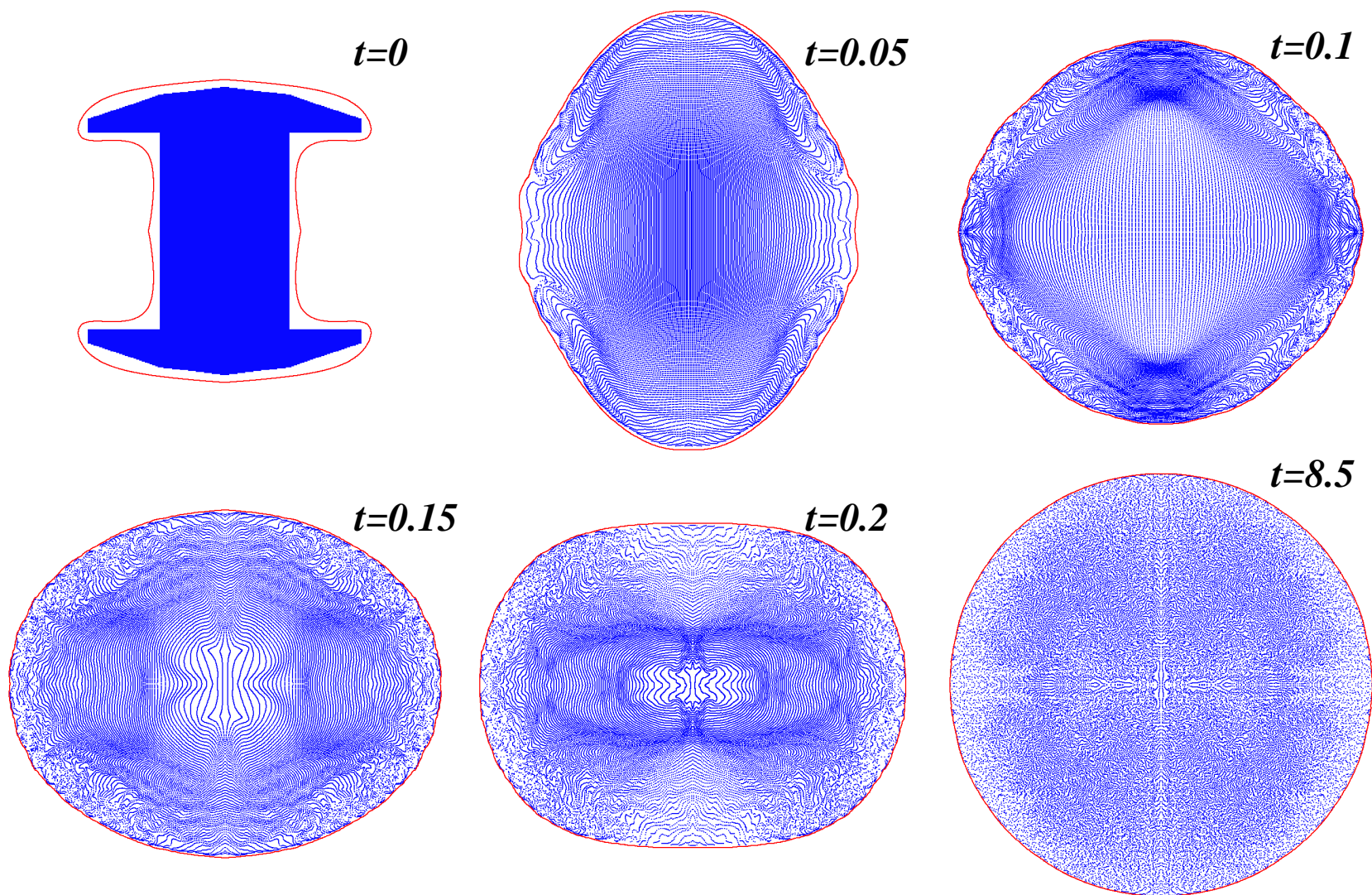


Figure 54. Material Point Position Plots for Simulation 4

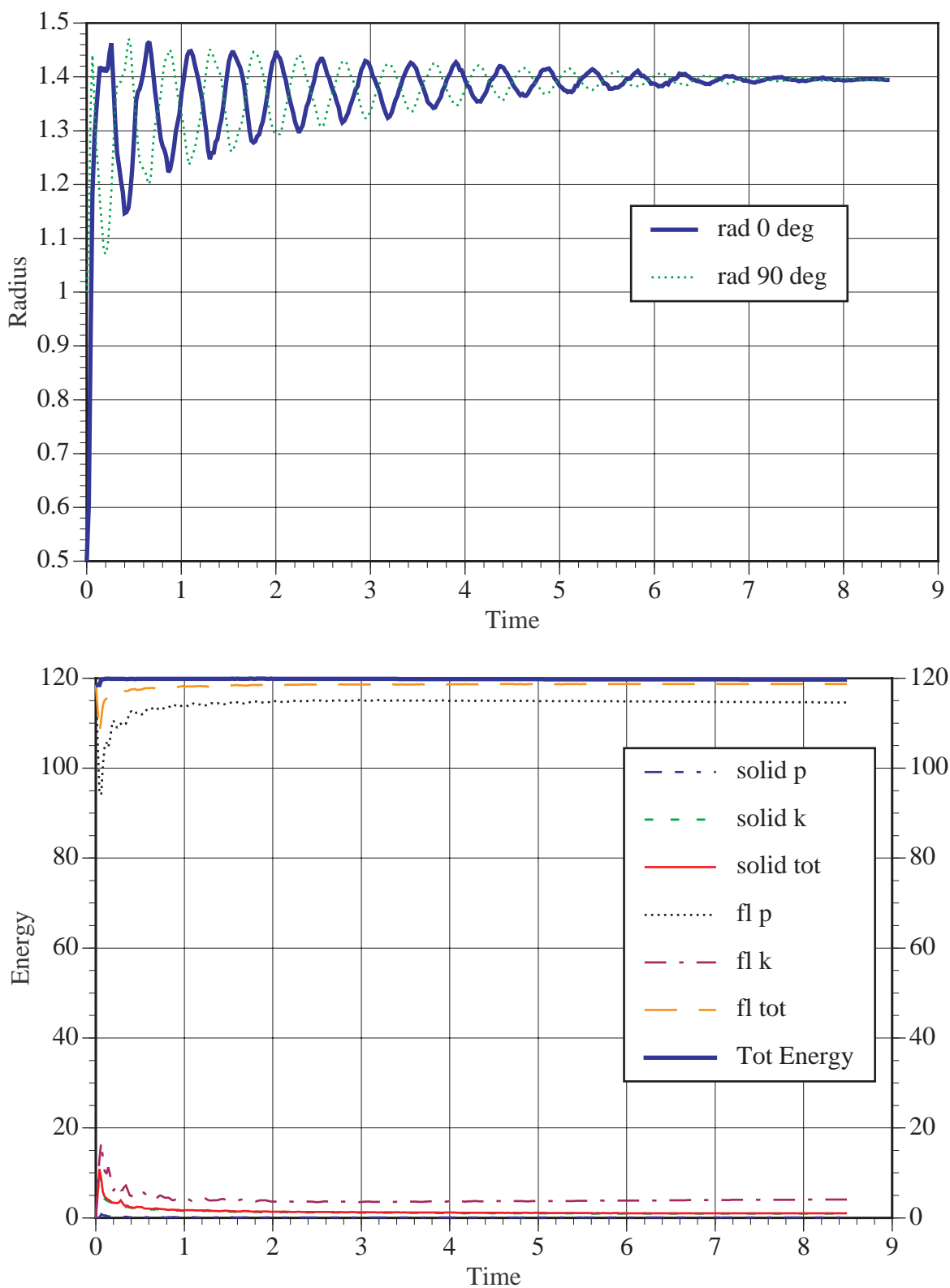


Figure 55. Radii and Energy for Membrane Expansion Simulation 4

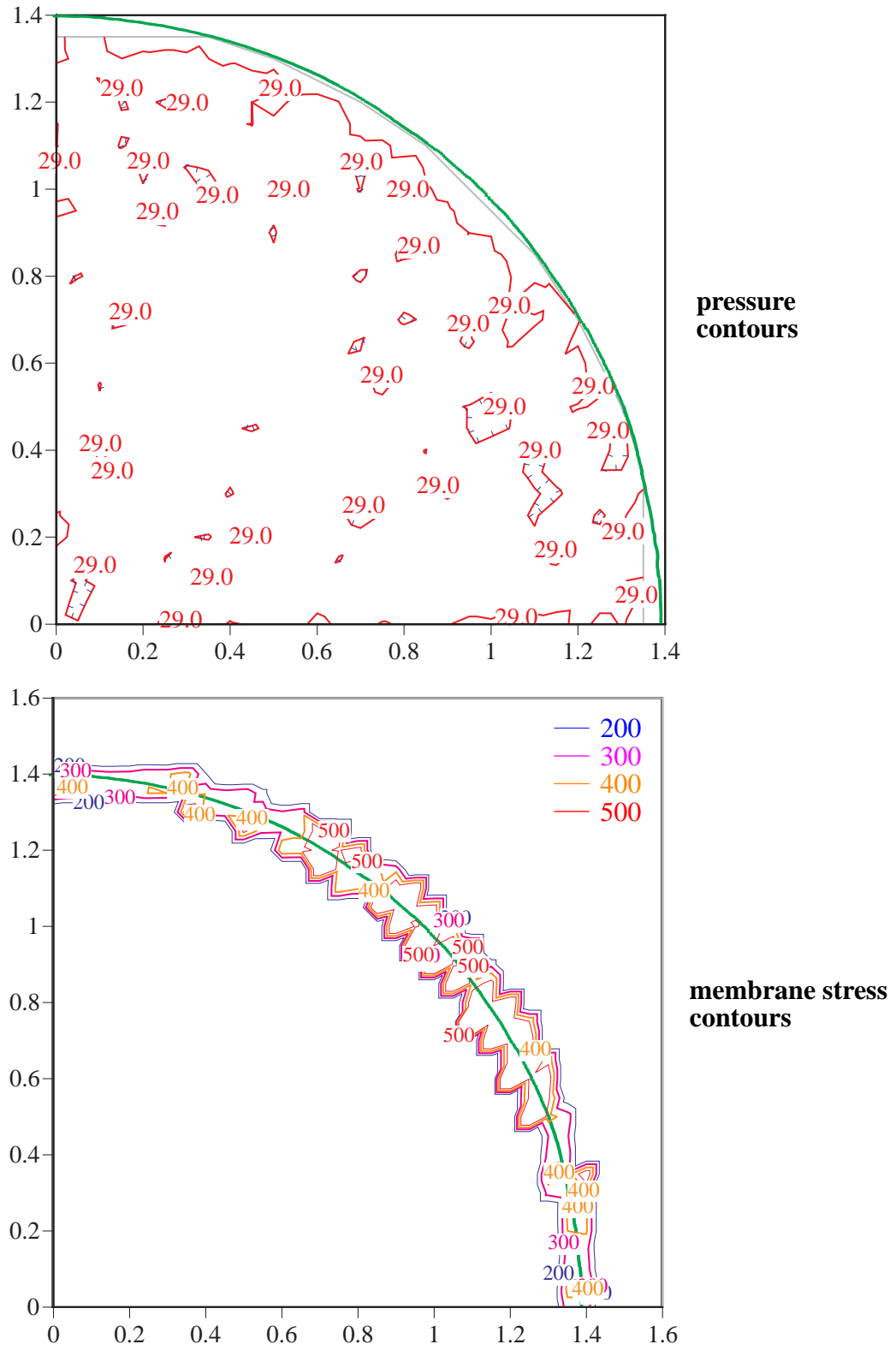


Figure 56. Pressure and Membrane Stress Contours for Simulation 4

6.3 Airbag Impact Simulation

6.3.1 Problem Description

Axisymmetric calculations of the interaction of an impacting probe with an inflated airbag were performed and are reported by de Coo (1989). The calculations were compared to experimental results. A schematic of the problem is shown in Fig. 57. The MPM is used to perform two different

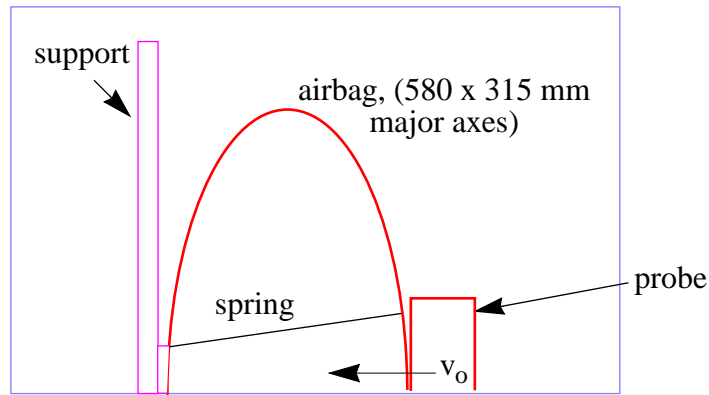


Figure 57. Airbag Impact Problem

simulations reported by de Coo where the diameter, mass, and impact velocity of the cylindrical probe were varied. The parameters of the two simulations are listed in Table 7.

Table 7: Airbag Impact Simulations

Simulation	Cylinder Diameter (mm)	Cylinder Mass (kg)	Cylinder Initial Velocity (m/s)
Cyl-200	200	6.03	3.90
Cyl-500	500	8.05	4.70

Material properties for the simulations are listed in Table 8.

Table 8: Airbag Impact Simulation - Material Properties

Airbag Property	Value	Air Property	Value
Thickness	0.5 mm	Specific Heat Ratio	1.4
Density	662 kg/m ³	Density	1.2156 kg/m ³

Table 8: Airbag Impact Simulation - Material Properties

Airbag Property	Value	Air Property	Value
Young's Modulus	$6 \times 10^7 \text{ N/m}^2$	Initial Pressure	4000 N/m^2
Poisson's Ratio	0.4	Initial Specific Internal Energy	8226 J/kg

Material properties of the cylinder were not listed. Therefore, Young's modulus was arbitrarily chosen to be $5 \times 10^7 \text{ N/m}^2$, and the density was adjusted to give the correct mass. The total airbag mass was derived to be 367.2 g. Thus, airbag material point volumes were prescribed such that the total airbag mass was correct.

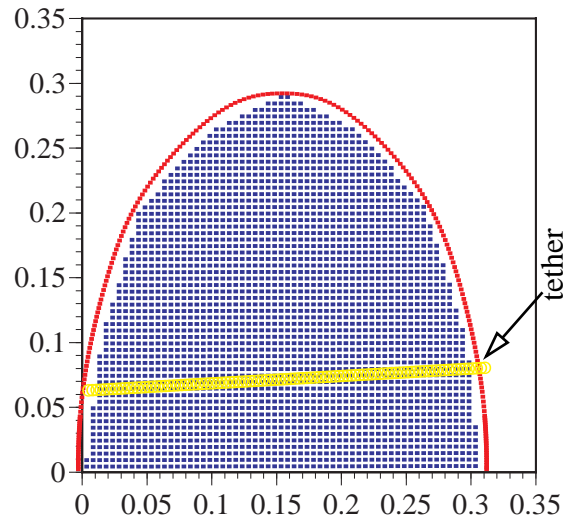
6.3.2 Cyl-200 Simulation Results

The initial positions of the membrane material points for this simulation were arrived at by conducting a simulation with a tether in the airbag as shown in Fig. 58. The objective of this exercise was to obtain an initial shape that matches that reported by deCoo. This simulation was run to approximately steady state where the gas expanded and put the tether in tension.

The plots of the probe and airbag material point positions for various times during the simulation are shown in Fig. 59. The effect of the using the tether to determine initial material point positions is seen in the first plot of Fig. 59. The tether pulled the material to the left (denoted by the arrow in the top left plot).

For the simulation with the probe, the tether was removed at the instant the simulation was started because in the axisymmetric MPM calculation this tether is actually a conical shape and affects the flow of the gas material points. The approximately 3,000 gas material points are not shown in Fig. 59 to increase the clarity of the airbag shape.

The predicted configuration of the airbag and impactor look plausible. The impactor causes the airbag to bulge out on the sides. It eventually stops and rebounds to the right.

**Figure 58. Initial Configuration with the Internal Tether**

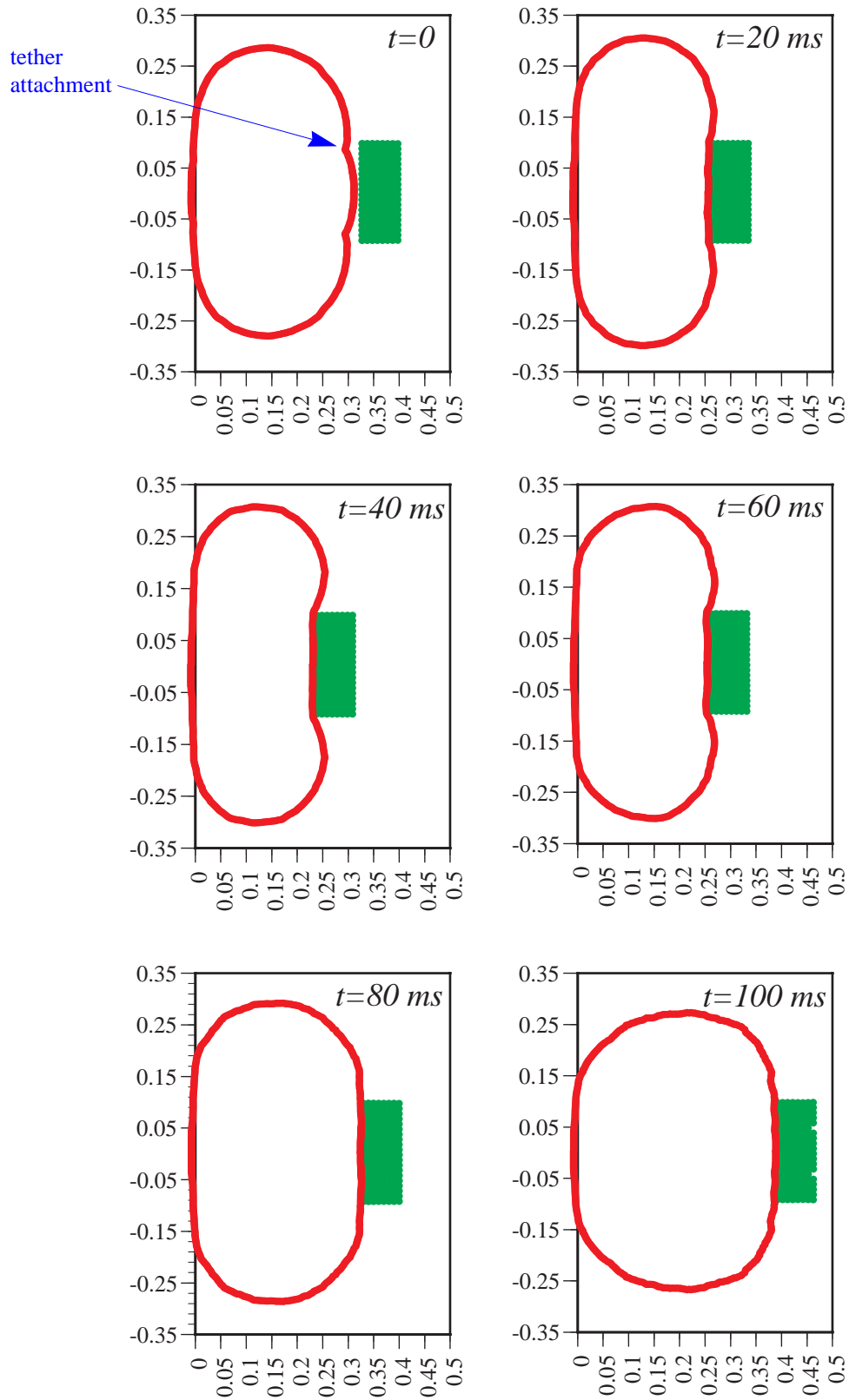


Figure 59. Deformed Airbag Shapes of Cyl-200 Simulation

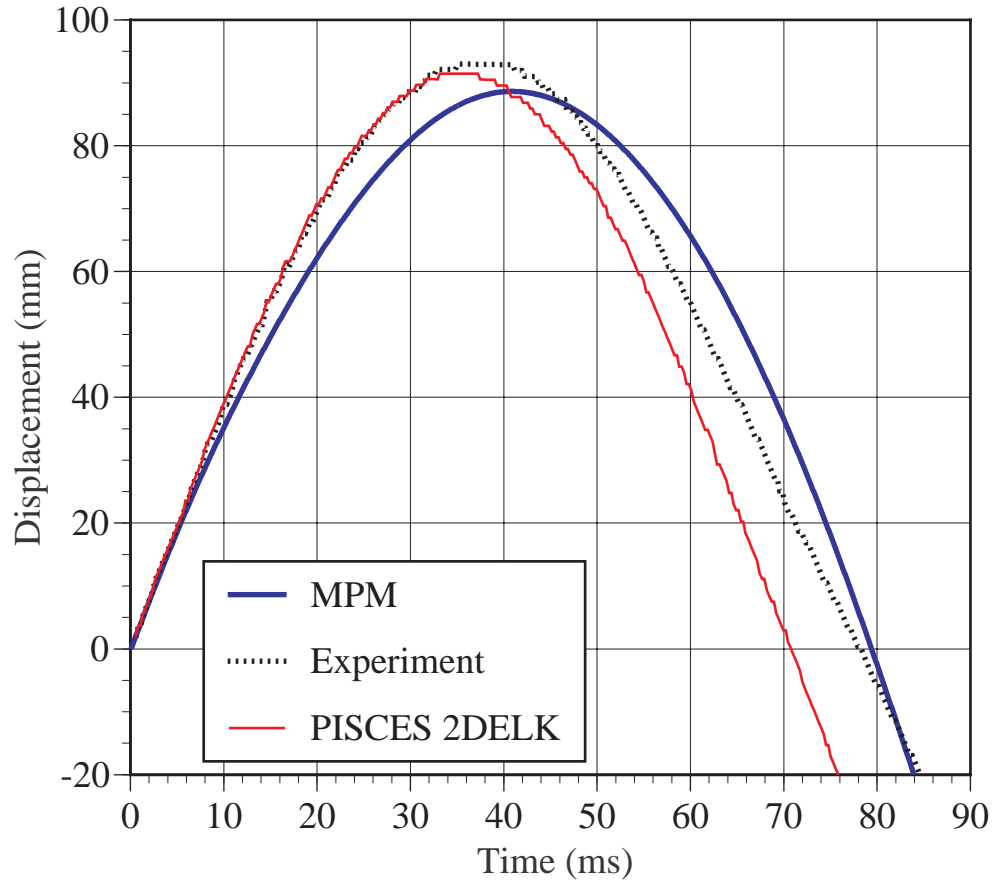


Figure 60. Displacement Results for Cyl-200 Simulation

Figure 60 shows a comparison of the displacement of the probe into the airbag for the MPM simulation, the PISCES simulation and the experiment. Figure 61 shows that the deformed airbag shapes at $t=40$ ms are similar.

The slope of the displacement past the peak appears different than the experimental slope. It was thought that the effect of no tether may cause a more rapid rebound of the impactor. To test this theory, a simulation was conducted where the tether was left in the simulation for the duration. It was observed that the slope past the peak closely matched that of the experiment. However, the cylinder displaced about 20 mm farther into the airbag. This excessive displacement is attributed to the “conical” effect of the tether on the gas flow for an axisymmetric calculation.

6.3.3 Cyl-500 Simulation Results

The deformed configurations at various times are shown in Fig. 62. While the “Cyl-200” simulation (Section 6.3.2) seemed somewhat insensitive to the initial configuration of the airbag, this simu-

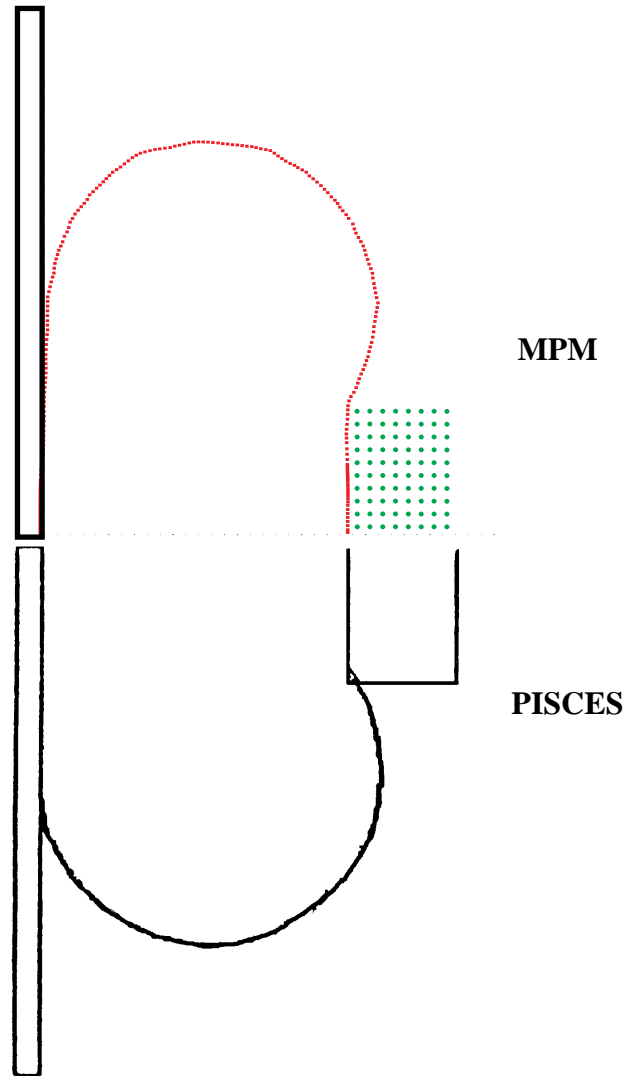


Figure 61. Comparison of PISCES and MPM Deformed Configuration for $t=40$ ms

lation was indeed dependent upon the initial shape. More effort was put forth in this simulation to get closer to the initial airbag shape used by de Coo. Preliminary simulations done with the 500 mm diameter cylinder and the airbag initial shape as shown in Fig. 58 (but without the tether) showed more displacement of the impactor into the airbag. The discrepancy seems to lie in the fact that when the airbag shape is flatter the initial force that decelerates the impactor is higher.

Figure 63 shows the displacements of the 500 mm diameter probe for the MPM and PISCES simulations and the experiment. It takes longer for the probe in the MPM simulation to rebound. A possible explanation for the difference comes from examining Fig. 64 that shows both simulation configurations at $t=40$ ms. There is more intimate contact between the probe and the airbag in the

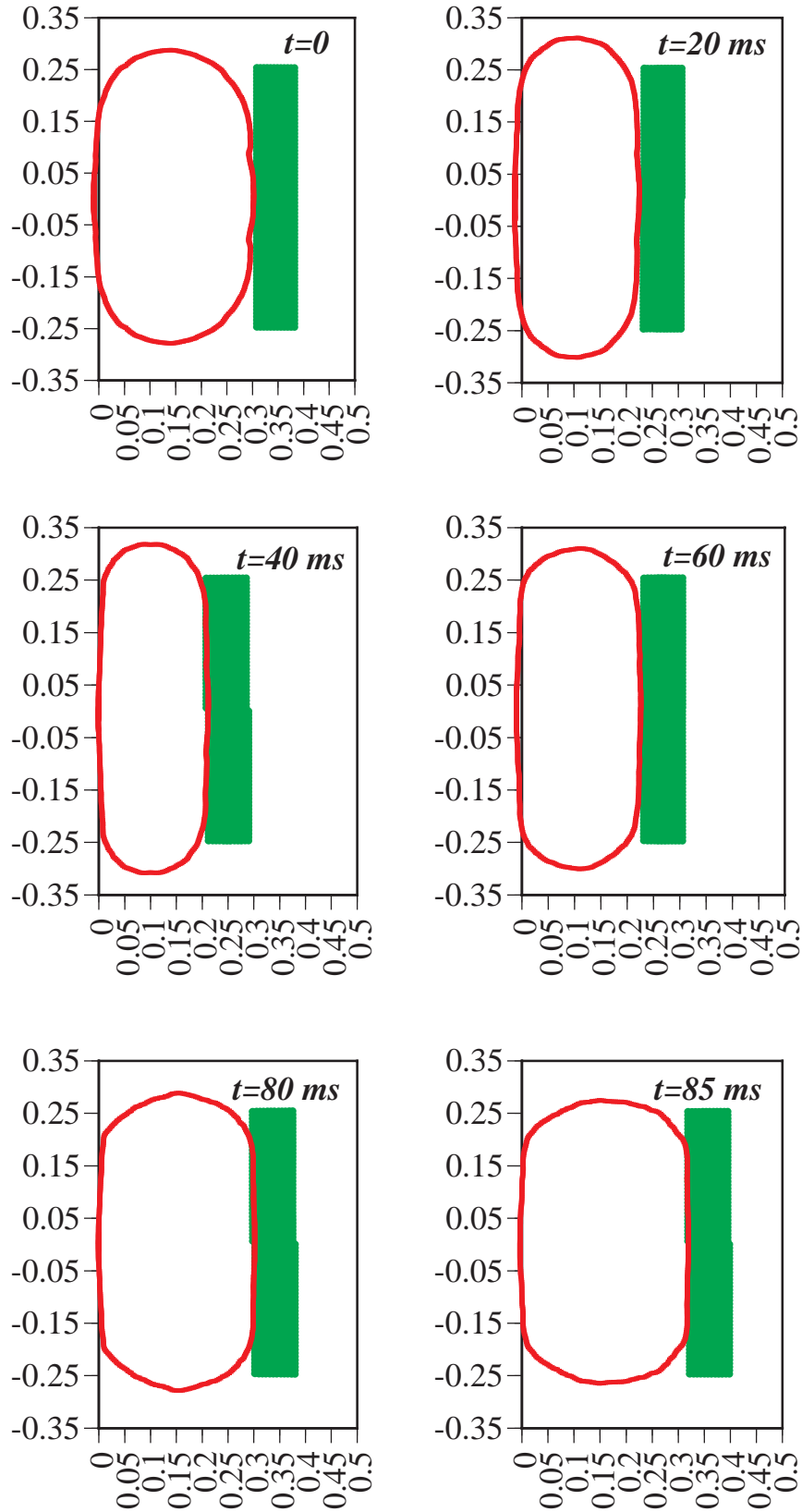


Figure 62. Deformed Airbag Shapes of Cyl-500 Simulation

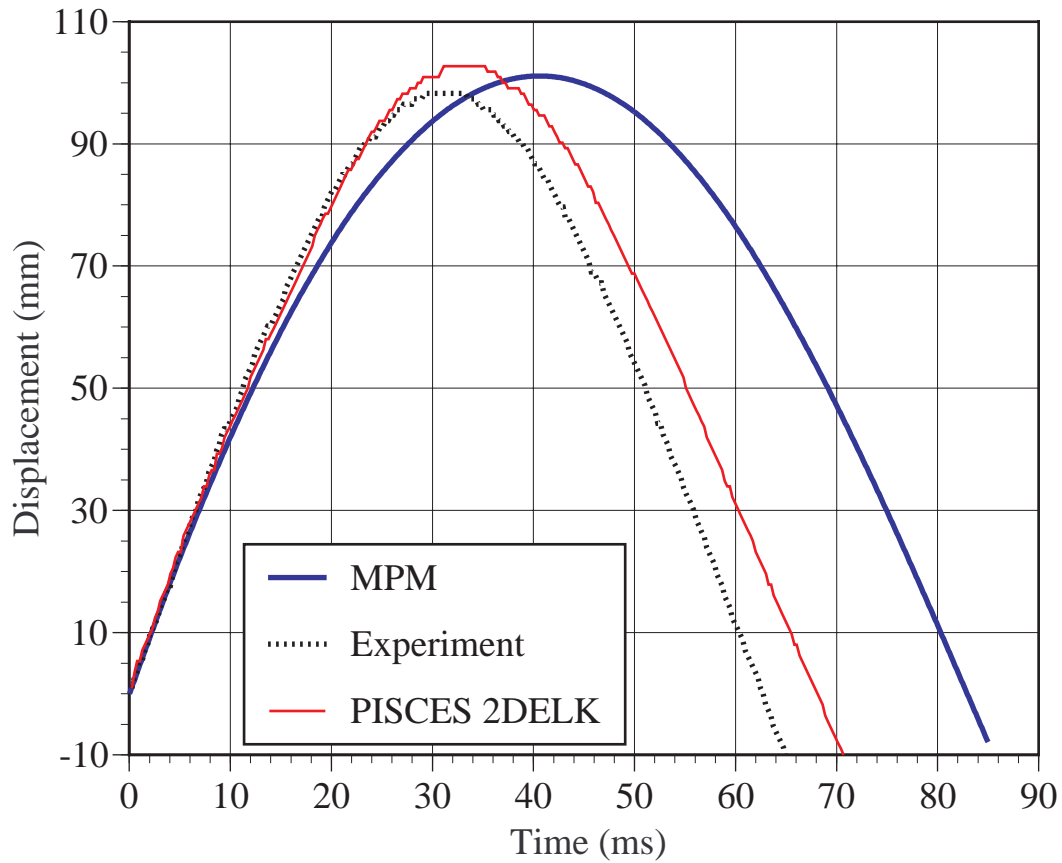


Figure 63. Displacement Results for Cyl-500 Simulation

PISCES simulation than in the MPM simulation. The explanation is that contact between materials in the MPM is defined on the grid. Thus, the probe can cause the airbag to displace away from the probe even when they are not in the same cell. To get the full effect of the gas pressure on the probe the airbag and probe should be very near to each other. This effect should be reduced with mesh refinement.

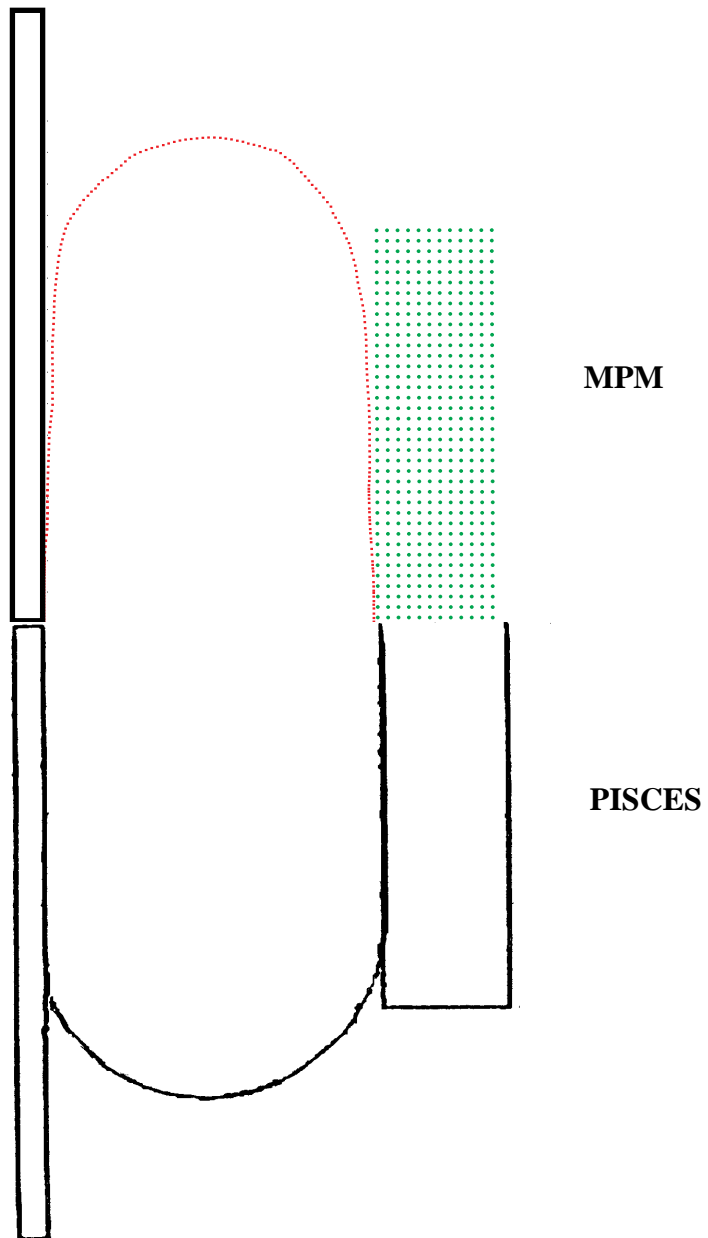


Figure 64. Comparison of PISCES and MPM Deformed Configuration for $t=40$ ms

6.4 Chapter 6 Summary

Simulations of three different problems demonstrate the ability of the MPM to model fluid-structure interaction. The interaction is achieved by tagging material points as either a fluid or a structure and applying the appropriate constitutive model to determine stress components of the material points. The standard MPM algorithm is then used to determine the grid-based internal force components from the stress components of each material point. The results of the problems simulated compare favorably to theory and/or experiment. Thus, the viability of a simple and robust algorithm for simulating the interaction of fluids and solids has been demonstrated.

CHAPTER 7. CALCULATING NORMALS

In all of the previously discussed problems, the local normal-tangential coordinate system for the membrane material points was determined using the connectivity of the points. This local coordinate system must be determined to evaluate the membrane constitutive equation. Figure 65 shows a membrane which is represented by the dashed lines through the material points which are solid circles. When using connectivity, the normal vector, \mathbf{n} , for a material point is determined from an average of its neighbors. For example the tangent angle with respect to horizontal for particle p in Fig. 65 is

$$\theta_p = \frac{\theta_{p-1,p} + \theta_{p,p+1}}{2}, \quad (7.1)$$

and the normal is 90 degrees offset from θ_p .

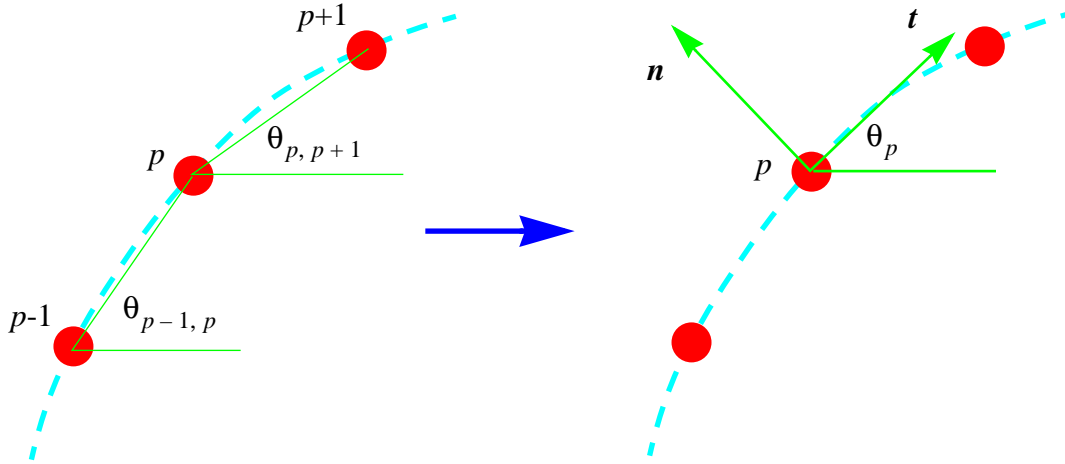


Figure 65. Determination of Normal to Material Point p

In two dimensions using the connectivity data to determine the local coordinate system is convenient and simple. In all the problems previously presented the connectivity data is obtained at no additional “cost” to the user since the connectivity is the same as the input sequence for the material points. However, a more general method to determine the local coordinate system would have the advantage of allowing the user more flexibility in specifying the initial shape of the membrane. This is especially true in a three-dimensional application. Using connectivity data in three-dimensions to determine the local coordinate system puts a heavy burden on the user to generate this data and will require additional storage in the code. This chapter presents four different methods for determining

the material point normals without using connectivity data. The methods are presented in order from the least effective to the most effective. Following the presentation of the various theories, each one is tested on a model problem.

7.1 Simple Color Function Approach

The basic idea with this method is to define a color (number) function that is constant along the membrane. The gradient of this function should be normal to the membrane. Thus, we start by defining a color, χ_p , for each membrane material point. All membrane material points have the same color.

Take the gradient of this function to define grid normals, \mathbf{n}_i , as

$$\mathbf{n}_i = \sum_p \mathbf{G}_{ip} \chi_p \quad (7.2)$$

where \mathbf{G}_{ip} are the gradients of the shape functions as defined in equation 3.14.

To obtain normals at the material points, interpolate the grid normals to the points using linear interpolation

$$\mathbf{n}_p = \sum_i \mathbf{n}_i N_i(\mathbf{X}_p). \quad (7.3)$$

This method averages the grid normals at neighboring grid points to define a normal at the material point position. The use of equation 7.2 to determine grid normals is not accurate because only material points in the membrane are used which is insufficient data to set up a gradient. The next two methods use higher order interpolation in an attempt to smooth the data.

7.2 Quadratic Interpolation

This method uses quadratic interpolation to define a cell color. The gradient of this cell color is defined at grid nodes. The material point normals are determined by interpolating from the grid gradient.

The cell color is defined as

$$\chi_c = \sum_p \chi_p S_c^{(2)}(\mathbf{X}_p) \quad (7.4)$$

where each material point will contribute to nine cell centers with the use of the weighting function $S_c^{(2)}(\mathbf{X}_p)$ (see Appendix 3.2 on page 174 for the definition of the weighting functions). The normal for grid node i is calculated by mapping the cell-centered color function to grid nodes using the same gradient operator as in equation 7.2

$$\mathbf{n}_i = \sum_i G_{ci} \chi_c \quad (7.5)$$

The material point normals are determined by equation 7.3.

7.3 Cubic Interpolation

In this method the grid color is defined by using cubic interpolation. The normals at the material points are calculated using a cubic gradient function. The higher order interpolation should give smoother varying results than the last method.

The grid color is defined as

$$\chi_i = \sum_p \chi_p S_c^{(3)}(\mathbf{X}_p) \quad (7.6)$$

where each material point will contribute to 16 grid vertices with the use of the weighting function $S_c^{(3)}(\mathbf{X}_p)$. Material point normals are determined using the cubic gradient operator as

$$\mathbf{n}_p = \sum_i G_{pi}^{(3)} \chi_i \quad (7.7)$$

7.4 Mass Matrix Approach

This method solves a system of equations simultaneously to determine grid colors that define a constant color along the membrane material points. We require that the grid colors, χ_i , define the constant material point color, χ_p^0 , according to

$$\chi_p^0 = \sum_i \chi_i N_i(\mathbf{X}_p) \quad (7.8)$$

where i ranges from one to the number of grid nodes. Now we ask what are the correct grid colors, χ_i ,

to use in determining the particle normals as

$$\mathbf{n}_p = \sum_i \chi_i \nabla N_i(\mathbf{X}_p). \quad (7.9)$$

Equation 7.8 will be manipulated to define a system of equations. Multiply both sides by $N_j(\mathbf{X}_p)$ and sum over p which gives

$$\sum_p \chi_p^0 N_j(\mathbf{X}_p) = \sum_p \chi_i \sum_i N_i(\mathbf{X}_p) N_j(\mathbf{X}_p). \quad (7.10)$$

Define a mass matrix T_{ij} as

$$T_{ij} = \sum_p N_i(\mathbf{X}_p) N_j(\mathbf{X}_p). \quad (7.11)$$

Now the system of equations to be solved can be written in matrix form as

$$\{\chi_p^0\}_{1 \times n_p} [N_{pj}]_{n_p \times n_g} = \{\chi_i\}_{1 \times n_g} [T_{ij}]_{n_g \times n_g} \quad (7.12)$$

where subscripts n_p , the number of material points, and n_g , the number of grid nodes, indicate the dimension of the vector or matrix. This approach can be applied globally or locally. That is, all membrane material points may be included in the solution to equation 7.12 to globally determine the grid colors or it may be applied on some subset of material points. A convenient choice of a subset of material points are those in a single computational cell. This reduces the mass matrix to 4 x 4 which can be quickly solved for the four grid colors.

If only membrane material points are included in a cell, one solution to the system defined in equation 7.12 is for each grid color to equal the membrane material point color. However, this is the trivial solution which results in no gradient of the color. The desired solution is obtained by adding one additional material point in the cell with a color different than that of the membrane color. When the system of equations is solved there is a gradient in the color, but the color is constant along the contour defined by the membrane material points.

7.5 Testing the Methods on a Model Problem

The model problem is that of a circular cylinder of radius 1.0 as shown in Fig. 66(a). The first test

of the methods is to calculate the normals for the first time step of the calculation before any deformation has occurred. In some cases the calculation is improved by adding material points interior to the membrane as in Fig. 66(b). In these cases four and 16 points-per-cell are used to illustrate the improvement. The second test is running the cylinder expansion problem using the different methods to calculate the normals.

7.5.1 Calculating Normals for the First Time Step

Figure 67 shows the results for the simple color function approach. For both a vacuum and 4 PPC fluid on the interior, the results are unacceptable. The results with fluid points on the interior as shown in Fig. 67(b) illustrate how the normals change direction at cell boundaries. This is a common prob-

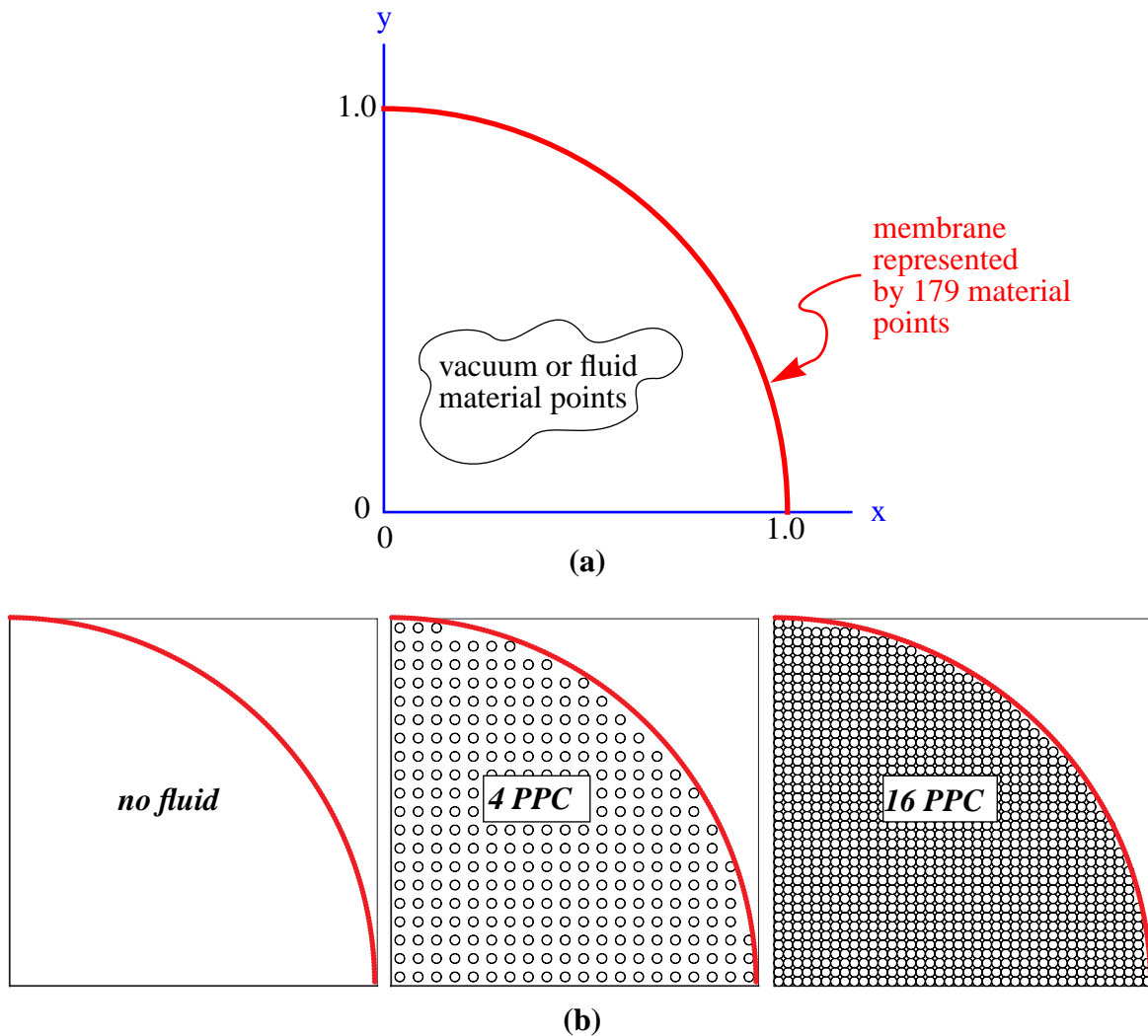


Figure 66. (a) Quarter Symmetry Model Problem for Testing Normal Calculations and (b) Material Point Plots for No Fluid, 4 Points-Per-Cell (PPC) Fluid, and 16 PPC Fluid

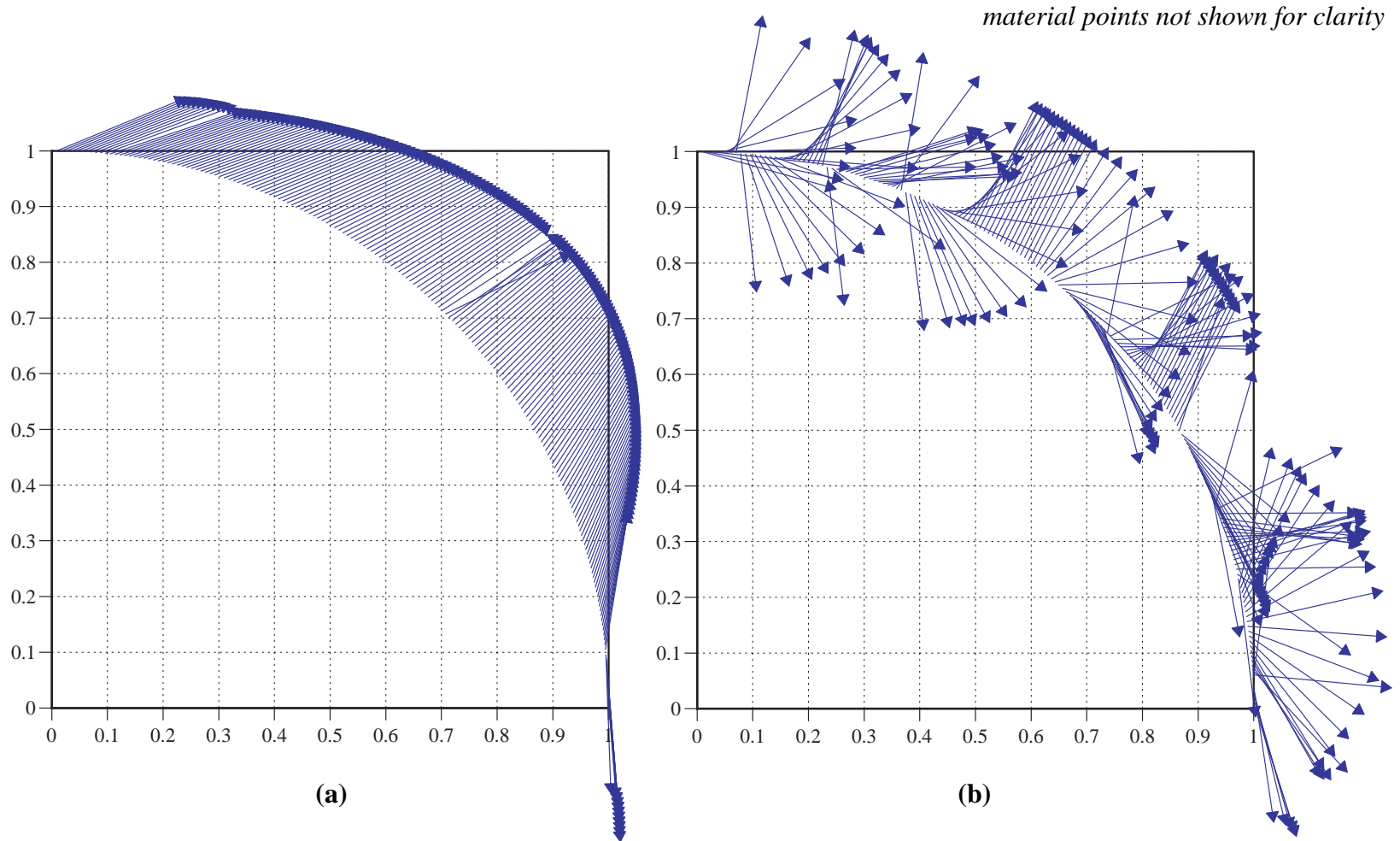


Figure 67. Normals to the Membrane Points Calculated by the Simple Color Function Method for: (a) Vacuum and (b) 4 PPC Fluid Material Points

lem that will also be seen in other methods.

Figures 68 and 69 show the results for the quadratic approach. Without interior fluid points the results are poor. However, with fluid points on the interior the results improve drastically. Further improvement is seen by having more fluid points on the interior as illustrated by comparing Fig.s 68(b) and 69. It should be noted that the normals only have to be determined to a precision of ± 180 degrees, although it is aesthetically pleasing to have the normals in a consistent direction.

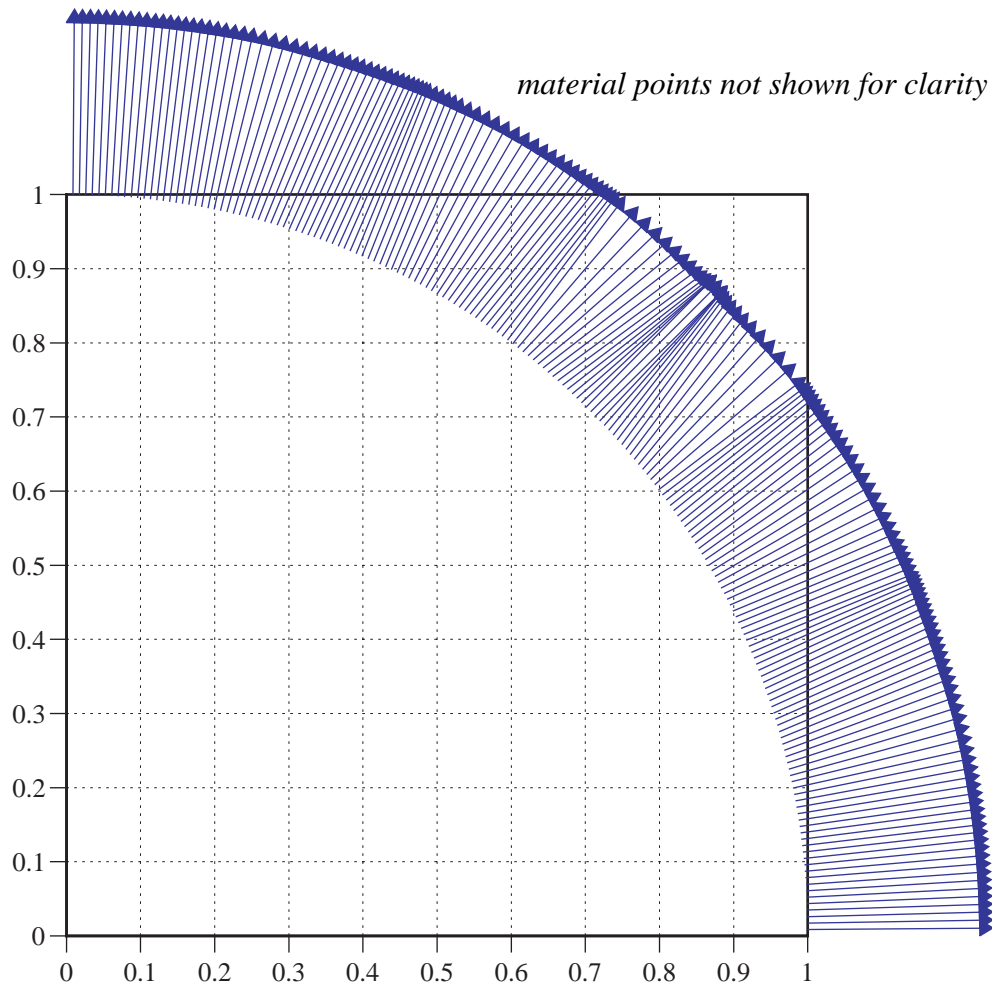


Figure 69. Normals to the Membrane Points Calculated by the Quadratic Method for 16 PPC Fluid Material Points

Figures 70 and 71 show the results for the cubic method. The results without fluid points on the interior are not as good as for the quadratic method, but with interior fluid points the results are better than with the quadratic method. As seen previously, the results improve with more interior fluid

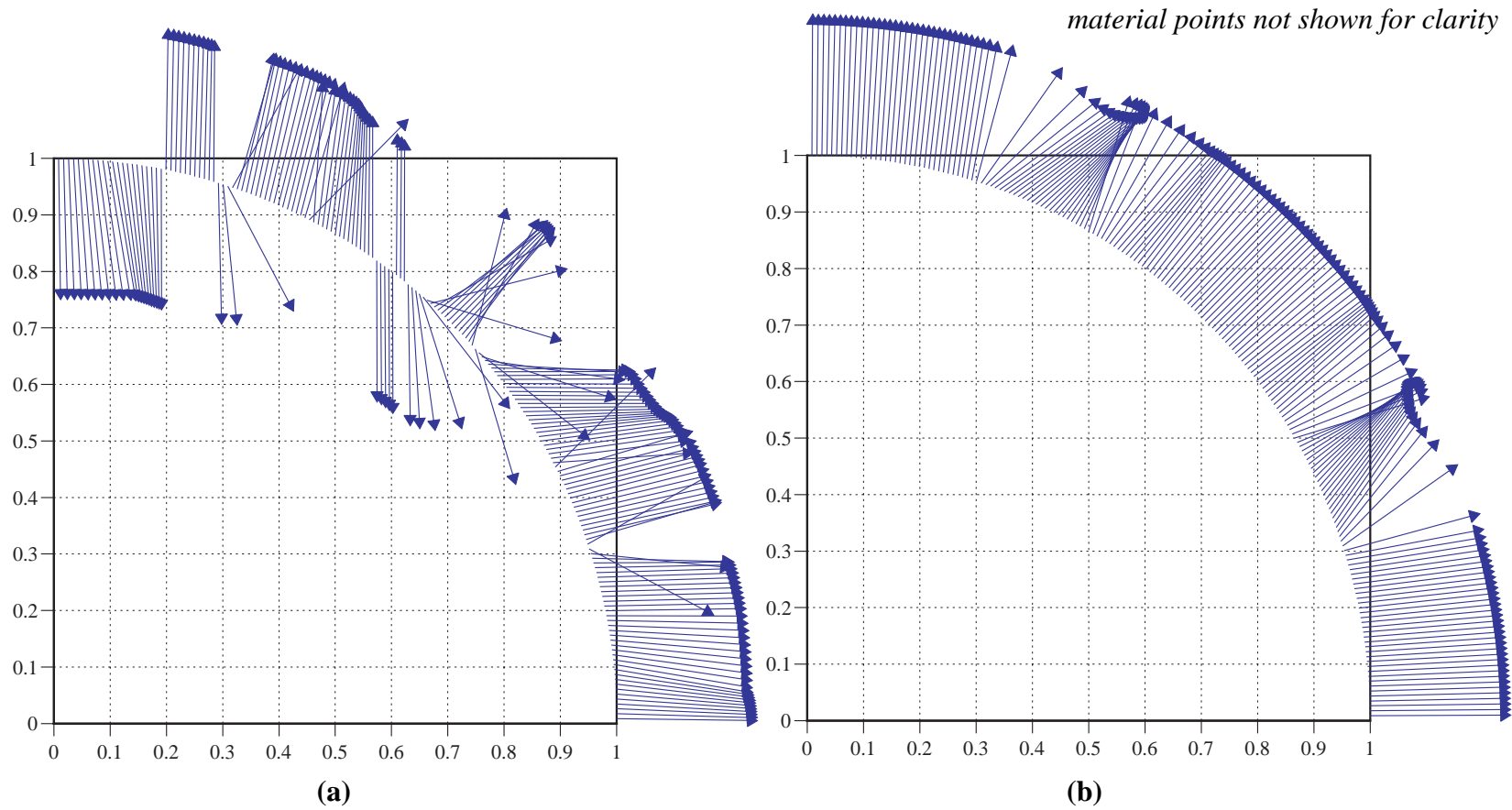


Figure 68. Normals to the Membrane Points Calculated by the Quadratic Method for: (a) Vacuum and (b) 4 PPC Fluid Material Points

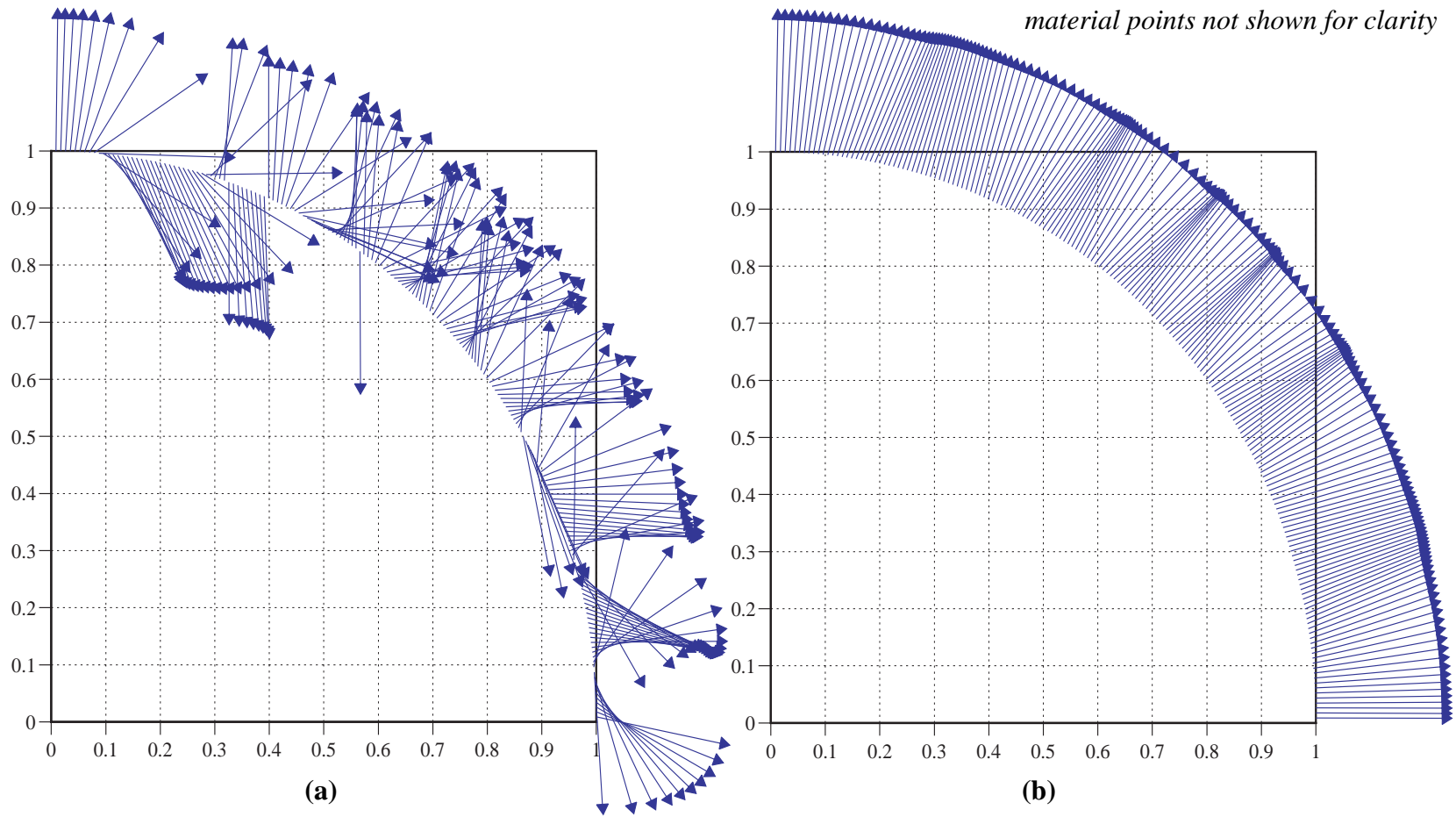


Figure 70. Normals to the Membrane Points Calculated by the Cubic Method for: (a) Vacuum and (b) 4 PPC Fluid Material Points

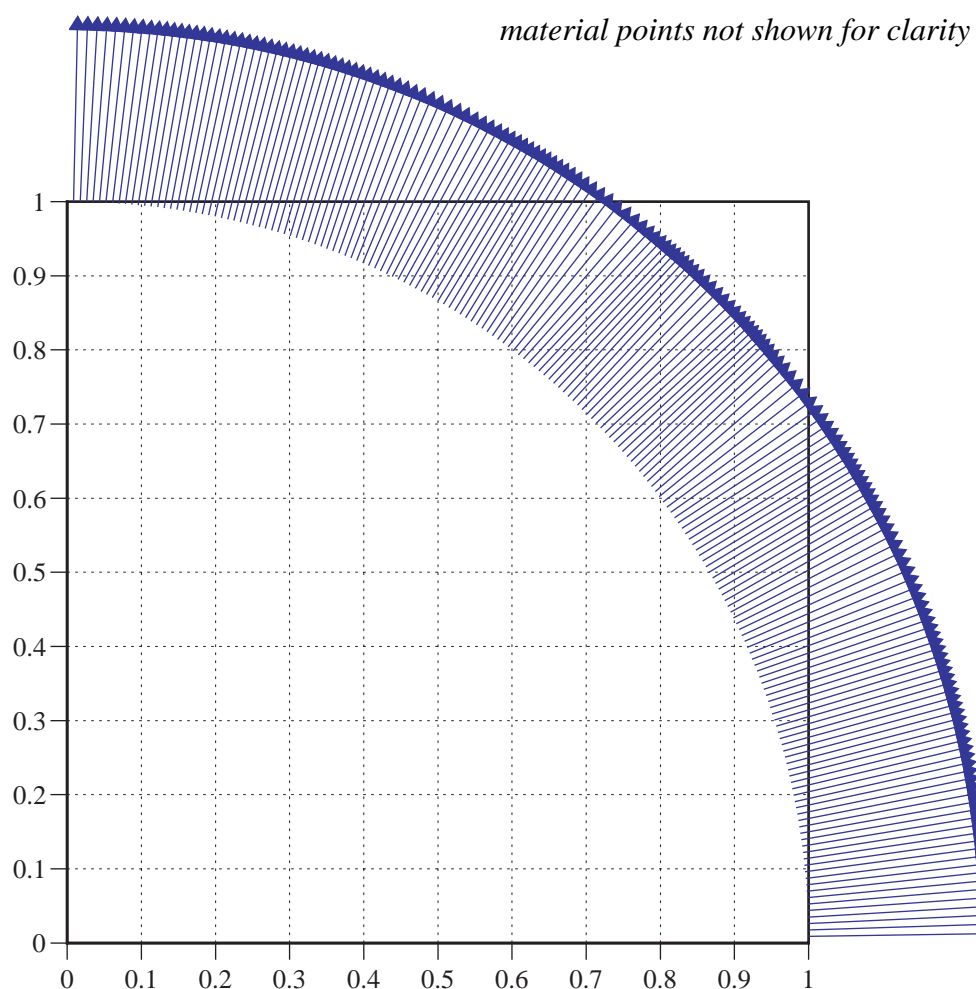


Figure 71. Normals to the Membrane Points Calculated by the Cubic Method for 16 PPC Fluid Material Points

points. The results for the cubic method were obtained by running a full 360 degree simulation because the boundary conditions for the cubic gradient were not implemented for this test.

The results for the mass matrix method are shown in [Fig. 72\(a\)](#). Here no fluid points are necessary for the interior. In fact the solution using the mass matrix method takes more computations if extra non-membrane material points are used, and the results are degraded. For comparison the normals using the material point connectivity are shown in [Fig. 72\(b\)](#). There is only a small difference between the mass matrix approach and the computation of normals using the connectivity of the material points.

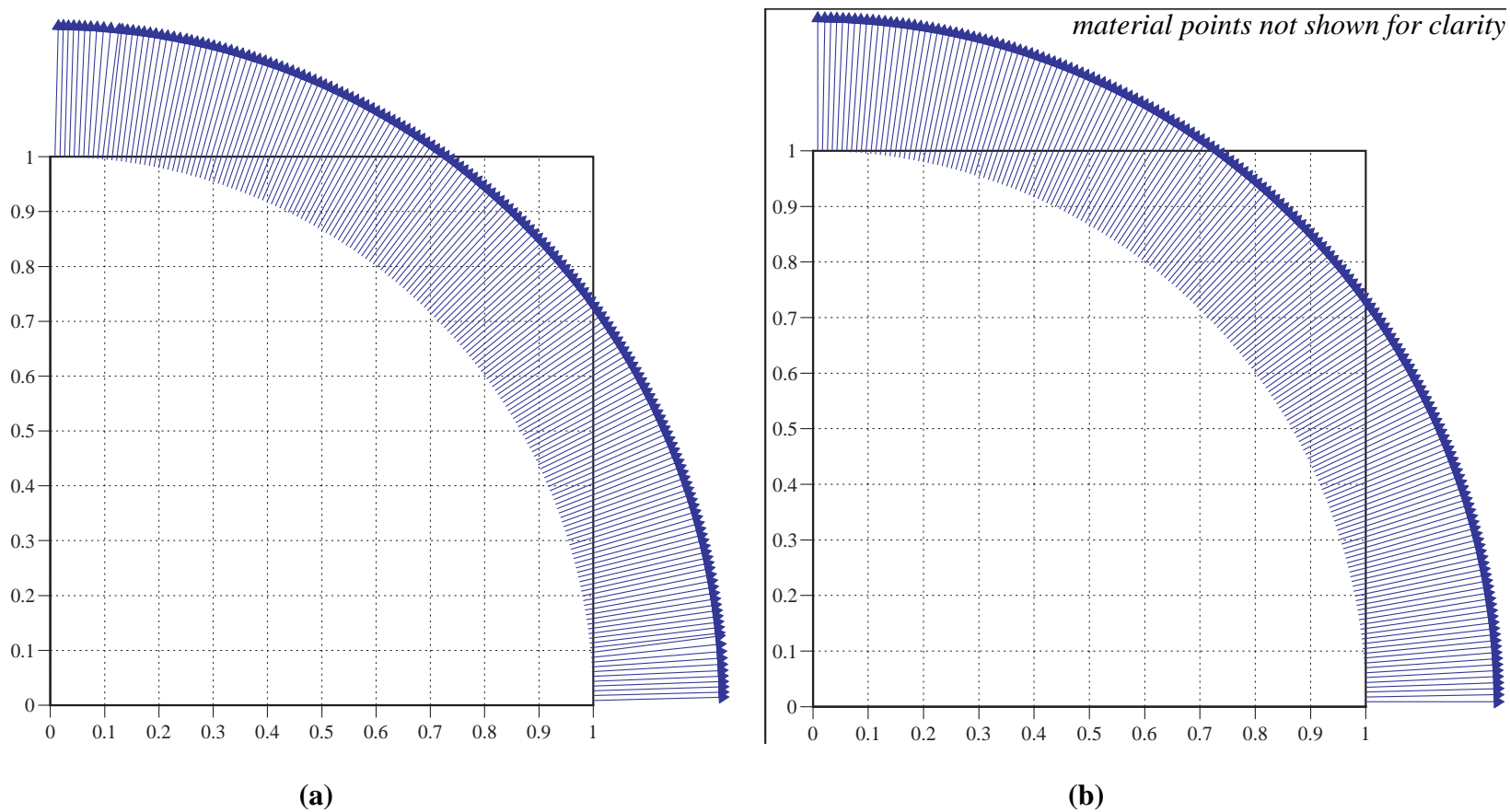


Figure 72. Normals to the Membrane Points Calculated by: (a) the Mass Matrix Method and (b) Using the Material Point Connectivity

7.5.2 Simulating an Expanding Cylindrical Membrane

The model problem was run with the internal fluid using the various methods to calculate the normals. The pressure of the fluid was such that the membrane should expand and begin to oscillate. The only method that is successful besides using the connectivity is the mass matrix method. Figure 73 shows the time history of the change in the average cylinder radius.* Using the standard connectivity

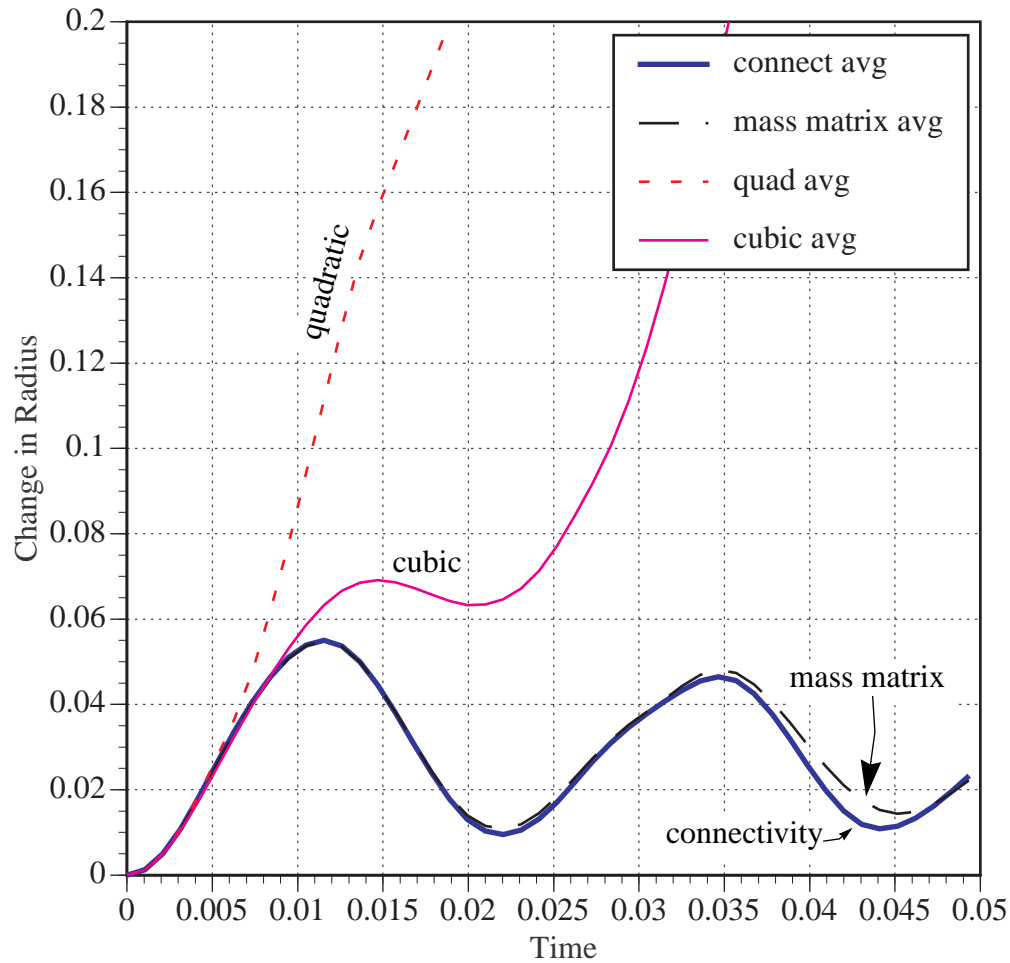


Figure 73. Change in Cylinder Radius

approach (thick solid line), the cylinder expands and then begins to oscillate. This is also true with the mass matrix approach (long dashed line). However, the other approaches are not successful. It can be seen in the cubic method (thin solid line) how the forces in the membrane only temporarily stop it from expanding. This indicates that the normals are not accurate enough to result in stresses directed

*The change in the radius of material points at 0, 45, and 90 degrees are averaged.

tangent to the membrane. The quadratic method is worse than the cubic method. By $t=0.005$ there are enough accumulated errors so that the membrane has virtually no resistance to expansion. The material point positions and normals associated with the membrane points at $t=0.01$ are shown in [Fig. 74](#) for the two unsuccessful methods - quadratic and cubic. The material point positions for the four methods at $t=0.05$ are shown in [Fig. 75](#). The boundary of the simulation was at $x,y=1.5$, and a zero velocity boundary condition was placed on these boundaries, so material will not expand beyond that point.

Figures [75\(a\)](#) and [75\(b\)](#) are similar, which is expected since the cylinder deflections were similar ([Fig. 73](#)). Figure [75\(c\)](#) shows results for the cubic method. Some forces in the membrane were generated to resist expansion. Figure [75\(d\)](#) shows results for the quadratic method where the membrane did not resist expansion much at all. The membrane has been pushed against the boundaries of the domain by the fluid.

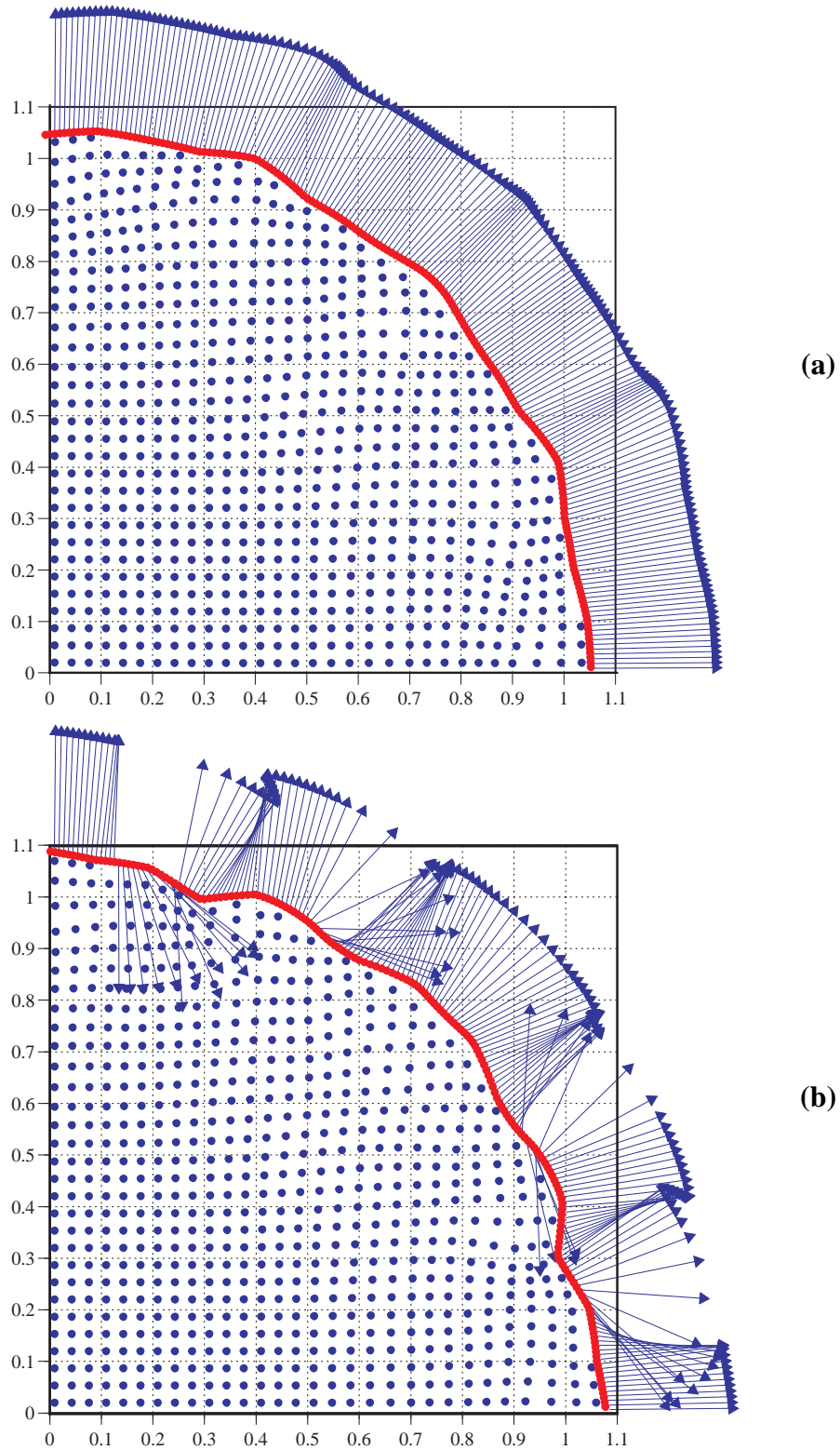


Figure 74. Material Point Positions and Normals for: (a) the Cubic Method and (b) the Quadratic Method at $t=0.01$

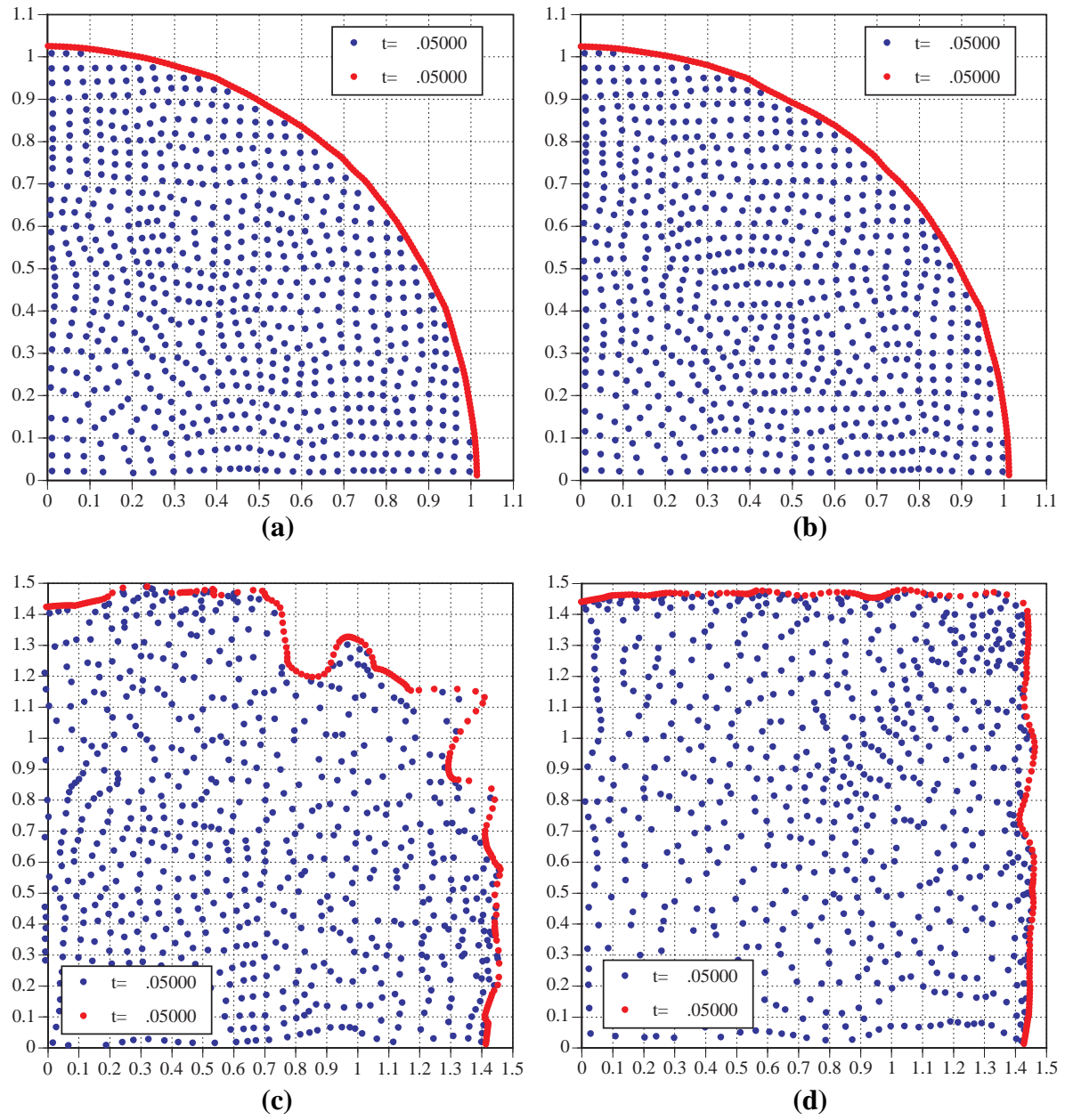


Figure 75. Material Point Positions When Calculating Normals Using (a) Connectivity, (b) Mass Matrix (linear shape functions), (c) Cubic, and (d) Quadratic Methods

7.6 Further Analysis of the Mass Matrix Method

This section quantifies the effectiveness and the limitations of the mass matrix method when using linear shape functions and solving a four-by-four system for each cell containing membrane points.

Recall that we desire that the grid colors, χ_i , define a constant material point color, χ_p^0 , according to

$$\chi_p^0 = \sum_i \chi_i N_i(\mathbf{X}_p) \quad (7.13)$$

where i ranges from one to the number of grid nodes. The particle normals are defined to be

$$\mathbf{n}_p = \sum_p \chi_i \nabla N_i(\mathbf{X}_p). \quad (7.14)$$

There will be some limitations in describing the particle normals with the gradient of linear shape functions. Since the shape functions are linear, the gradients with respect to x will be constant in the x direction and vary linearly in the y direction. Similarly, the gradients with respect to y will be constant in the y direction and vary linearly in the x direction. Thus, the polynomial describing gradients is only linearly complete. The gradients of linear membrane contours should be accurately defined. Other nonlinear contours may not be accurately represented. However, there is x - y coupling that will be able to capture some nonlinearities in the membrane color contour.

Figure 76 shows results for a linear membrane contour oriented at 45 degrees to horizontal. The membrane points are represented by solid circles. The color of the membrane material points is 1.0, and the contours of color throughout the cell are plotted with solid black lines. These contours are determined using the right-hand-side of equation 7.13 and the calculated grid colors χ_i . As expected, the line connecting the membrane points represents a constant contour of 1.0. The predicted normal vectors are exactly at 90 degrees (to machine precision) to the line connecting the membrane points. Figure 77 shows results for a linear membrane contour oriented at 80 degrees from horizontal. The material points are not evenly spaced. Again, the color contour of 1.0 is aligned with the material points. The normal vectors are close to the theoretical angle. On average, each vector orientation is within 0.01 degrees of being exact.

Figure 78 shows results for a nonlinear configuration of the membrane points. Here the function defining the location of the material points is $xy + 0.1x + C = 0$, where C is a constant. Note the

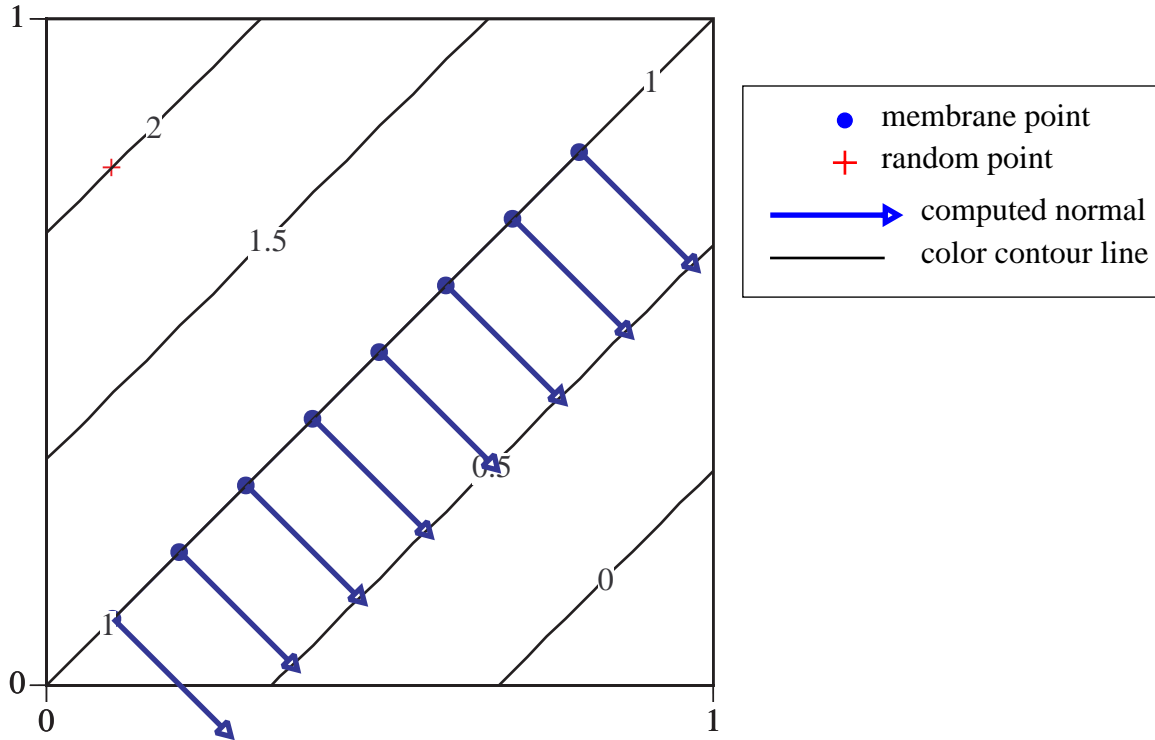


Figure 76. Membrane Points Oriented at 45 deg, Associated Color Contours, and Calculated Normals

presence of the bilinear term xy . It can be seen in [Figure 78](#) that the color contour of 1.0 is aligned with the material points. The endpoints (x,y) of the predicted normal vectors are aligned with the theoretical ones to within the fifth decimal place.

[Figure 79](#) shows the results for a circular contour of membrane particles of radius 0.4. In this case, the color contour of 1.0 does not lie on membrane points. The interpolated color of the membrane points ranges from about 1.08 to 1.25. As expected the normals are not as accurate as in previous cases. One material point normal is off several degrees while others are more accurate (compare solid thick vectors to dashed vectors).

The previous examples show that the method can give very good results. However, there are certain conditions under which this method fails. If the particles are arranged in a cell such that the mass matrix is singular or nearly singular, the resulting solution for grid color is inaccurate. For example consider a vertical arrangement of membrane material points. In this case the mass matrix is nearly singular. The arrangement of material points and results are shown in [Figure 80\(a\)](#). The normals, which should be exactly horizontal, are obviously inaccurate. If one material point is perturbed in the

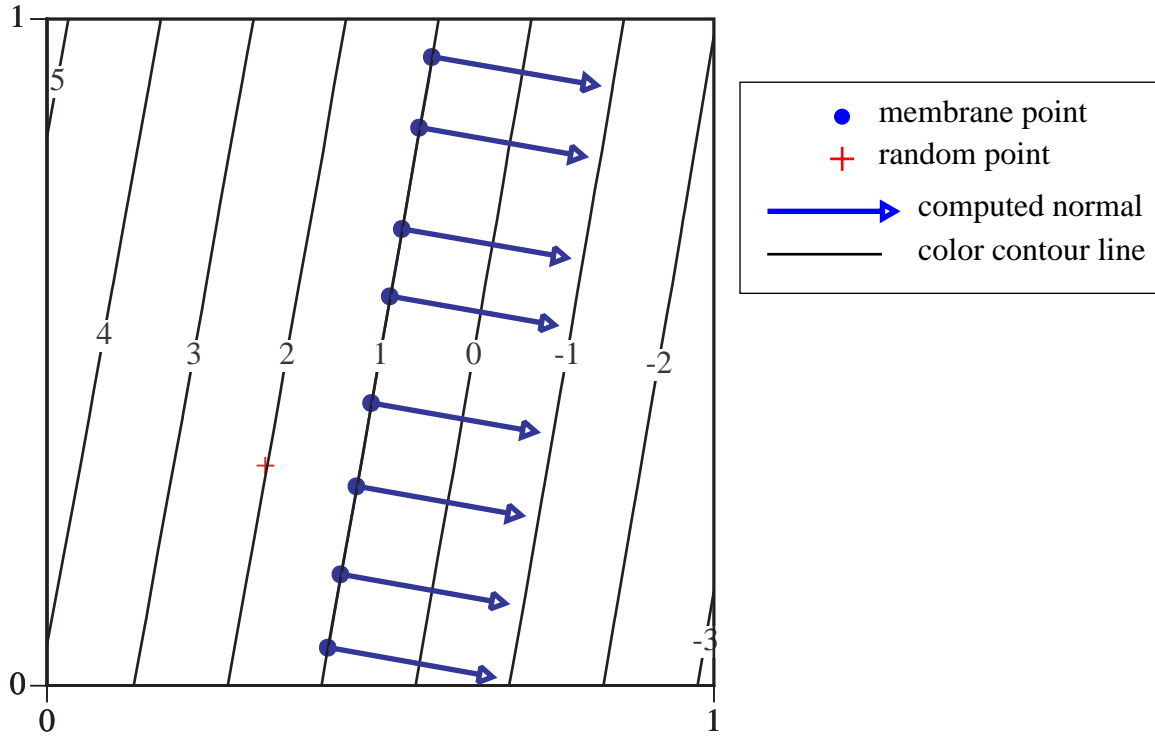


Figure 77. Membrane Points Oriented at 10 deg, Associated Color Contours, and Exact Calculated Normals

x direction, the condition number of the mass matrix improves greatly. However, the solution for grid color is such that there is no gradient in the center of the cell as shown in [Figure 80\(b\)](#). In this case the resulting contours within the cell are nearly symmetric. The material points away from the cell center have accurate normals, but the one point near the center does not.

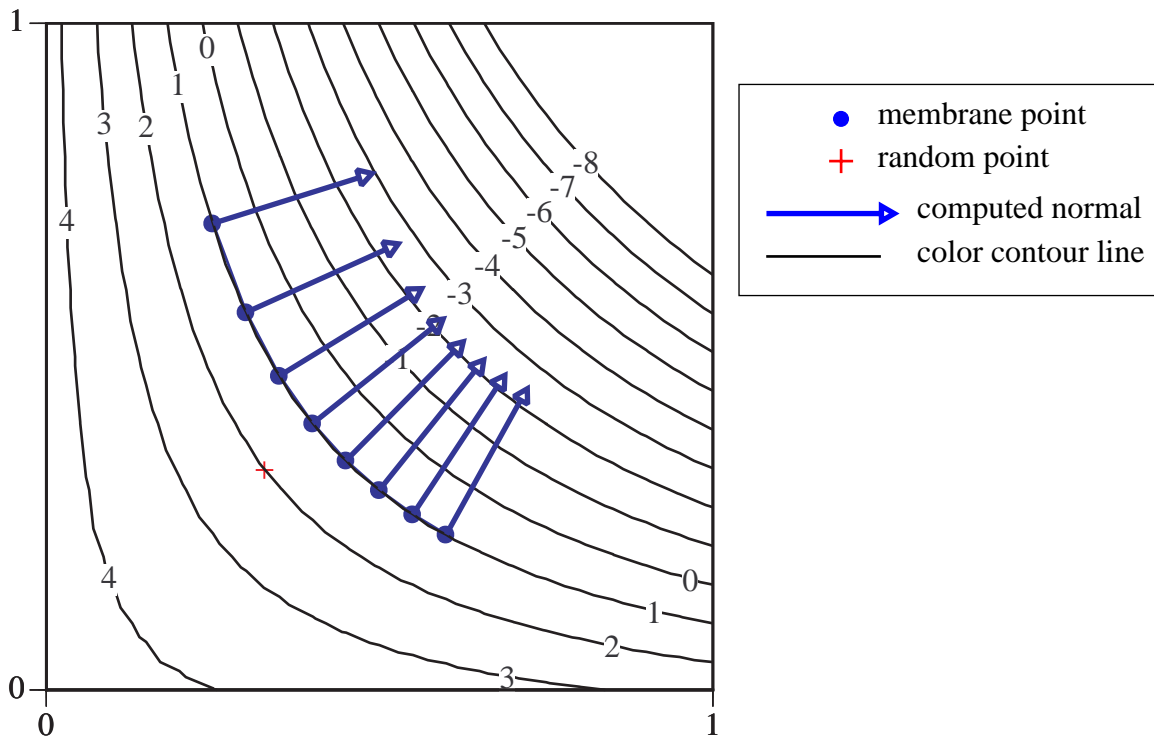


Figure 78. Membrane Points Along a Curve $xy=C$, Associated Color Contours, and Calculated Normals

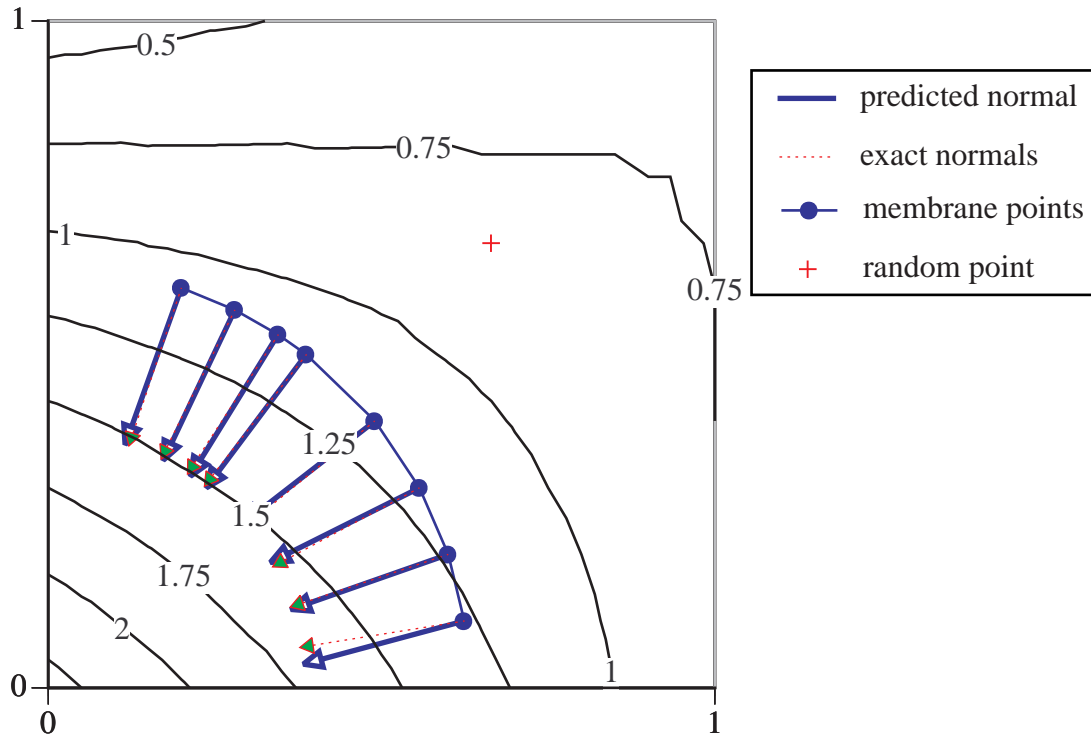


Figure 79. Membrane Points Along a Circular Arc and Associated Color Contours, and Exact and Calculated Normals

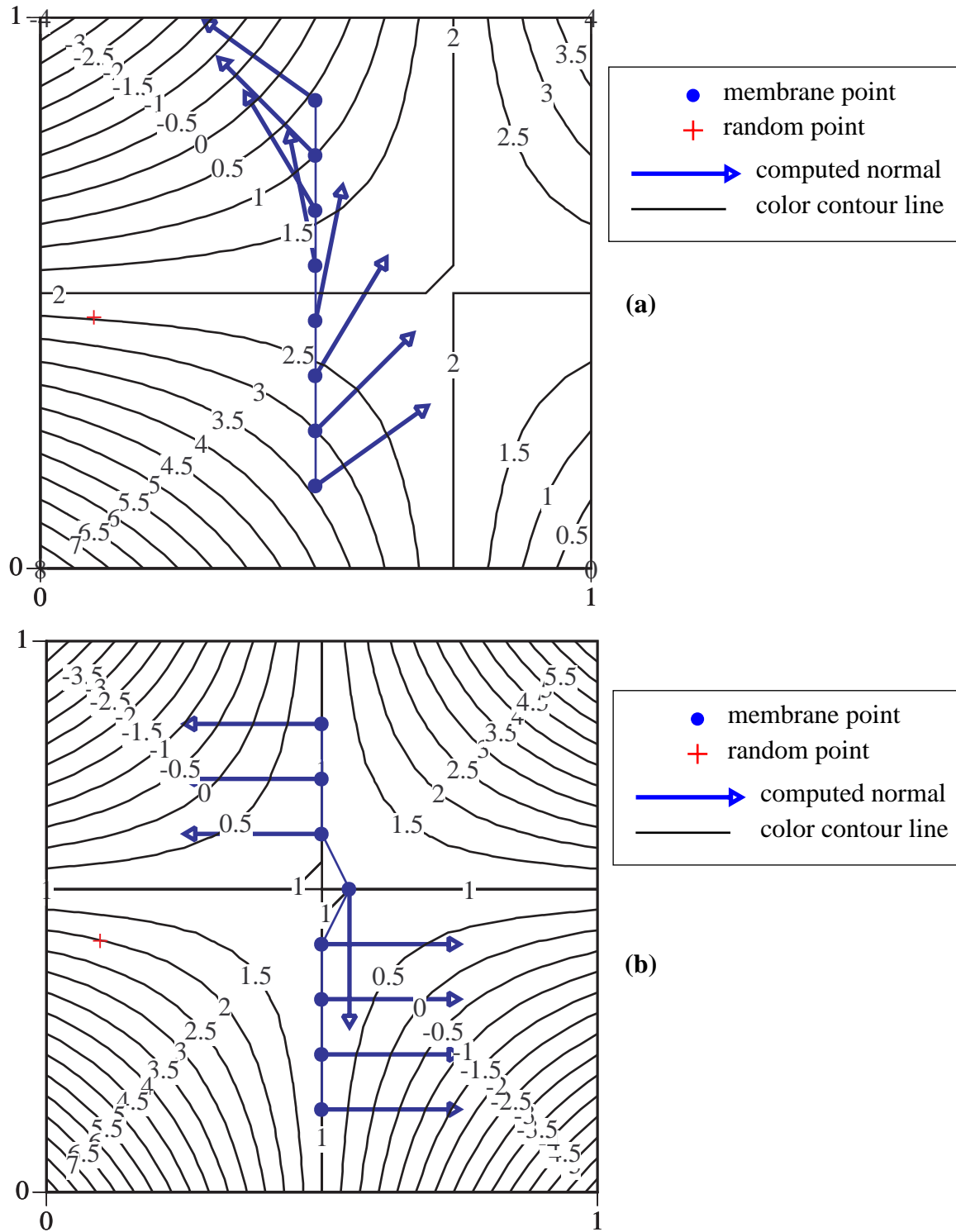


Figure 80. Cases Where the Method to Breaks Down: (a) Nearly Singular Mass Matrix, and (b) Nearly Symmetric Grid Color Solution that Results in No Gradient at the Cell Center

It can be shown that equation 7.14 can be written in terms of the material point local coordinates (r,s) and constants a and b as

$$n_{px} = \frac{s}{dx}a + b \quad \text{and} \quad n_{py} = -\frac{(1-r)}{dy}a + c \quad (7.15)$$

where constants a , b , and c are functions of the grid colors

$$\begin{aligned} a &= \chi_1 - \chi_2 + \chi_3 - \chi_4 \\ b &= -\chi_1 + \chi_2 \\ c &= -\chi_2 + \chi_3 \end{aligned} \quad (7.16)$$

The nomenclature for the local coordinates and grid colors is shown in Fig. 81. The relationship in

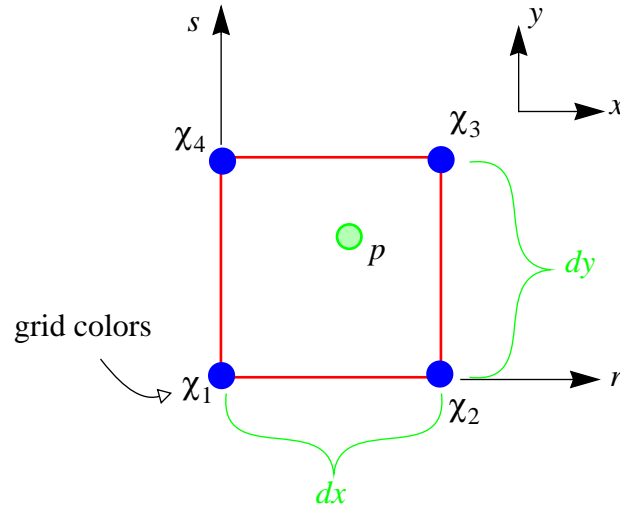


Figure 81. Nomenclature

equation 7.15 reveals a limitation in describing material point normals within a single cell. Consider three material points in a unit cell as shown in Fig. 82. Assume that the normals for the material points are as shown, and treat the unknowns as the constants a and b in equation 7.16. The equations for material points one and two are

$$\begin{aligned} s_1 a + b &= n_{px1} \\ s_2 a + b &= n_{px2} \end{aligned} \quad (7.17)$$

where s_1 and s_2 are the local y-coordinate of the two material points, and n_{px1} and n_{px2} are the x components of the particle normals. Now assume material points one and two have the same y-coordi-

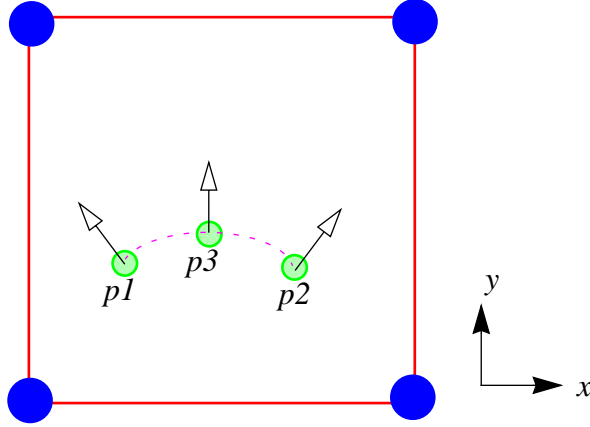


Figure 82. Unit Cell with Three Material Points

nate. The two equations now have the same left-hand-side. Thus, the only solution is for the normal component in the x direction to be the same for both material points which, in general, is inconsistent with our initial assumptions about the normal vectors. A similar analysis can be done for material points that have the same x -coordinate. The result in this case requires that the normal component in the y -direction be the same for the two particles. It is difficult to infer the performance of this method given the above discussion because when one assumes the material points have an identical x or y local coordinate nothing is said about the other. The following example helps to quantify the problems associated with predicting normals to an arc within a cell.

Figure 83 shows the color contours and normals to a circular arc. The lower plot of Figure 83 and those in Figure 84 show the normals after progressively rotating the circular arc about the center of the cell. It can be seen that material points that have nearly the same y coordinate (near the top of the arc) have x components of normals that are nearly the same. This is most evident in the lower plot of Fig. 84 where the material points with identical y coordinates are labeled.

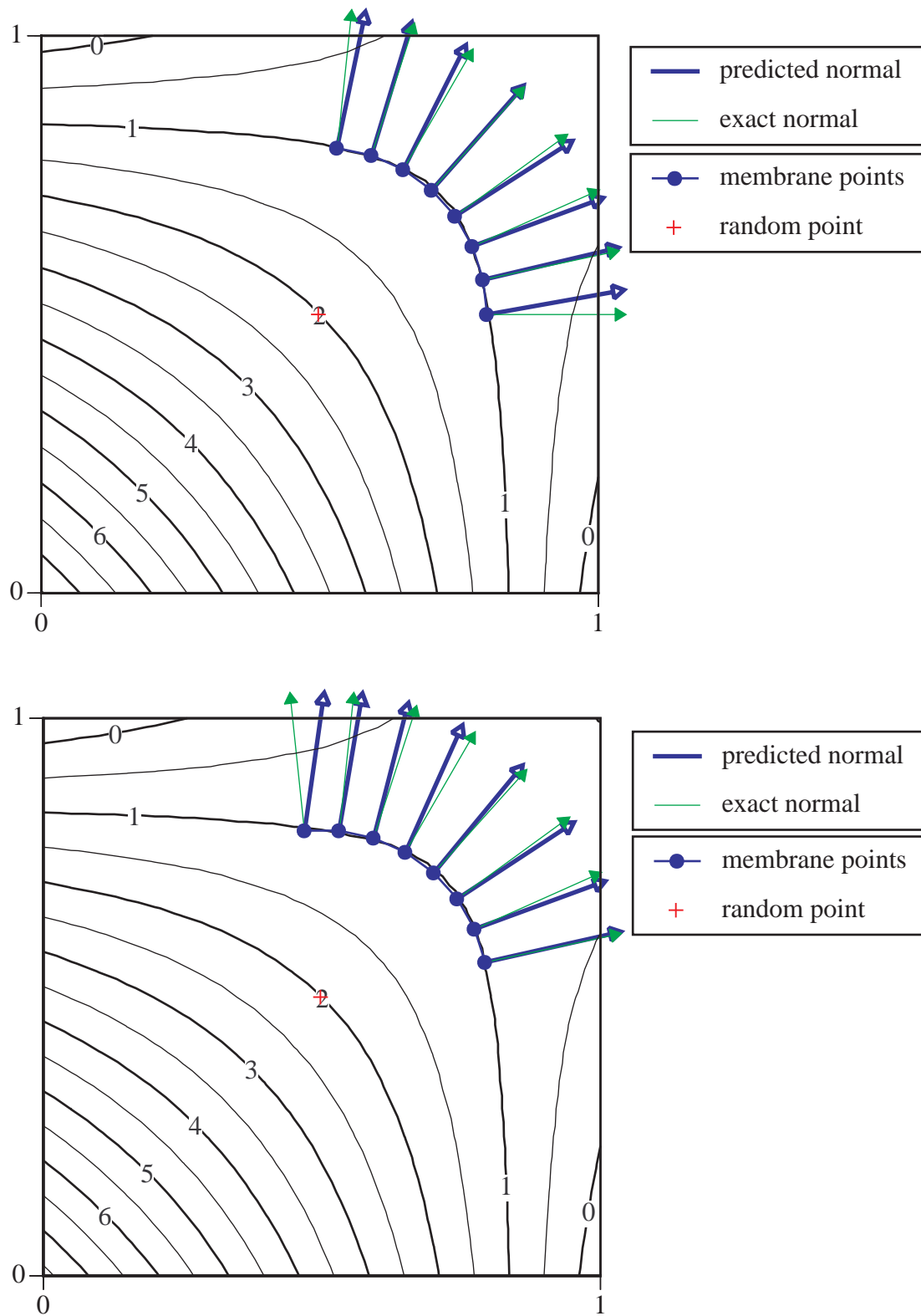


Figure 83. Normals to a Circular Arc

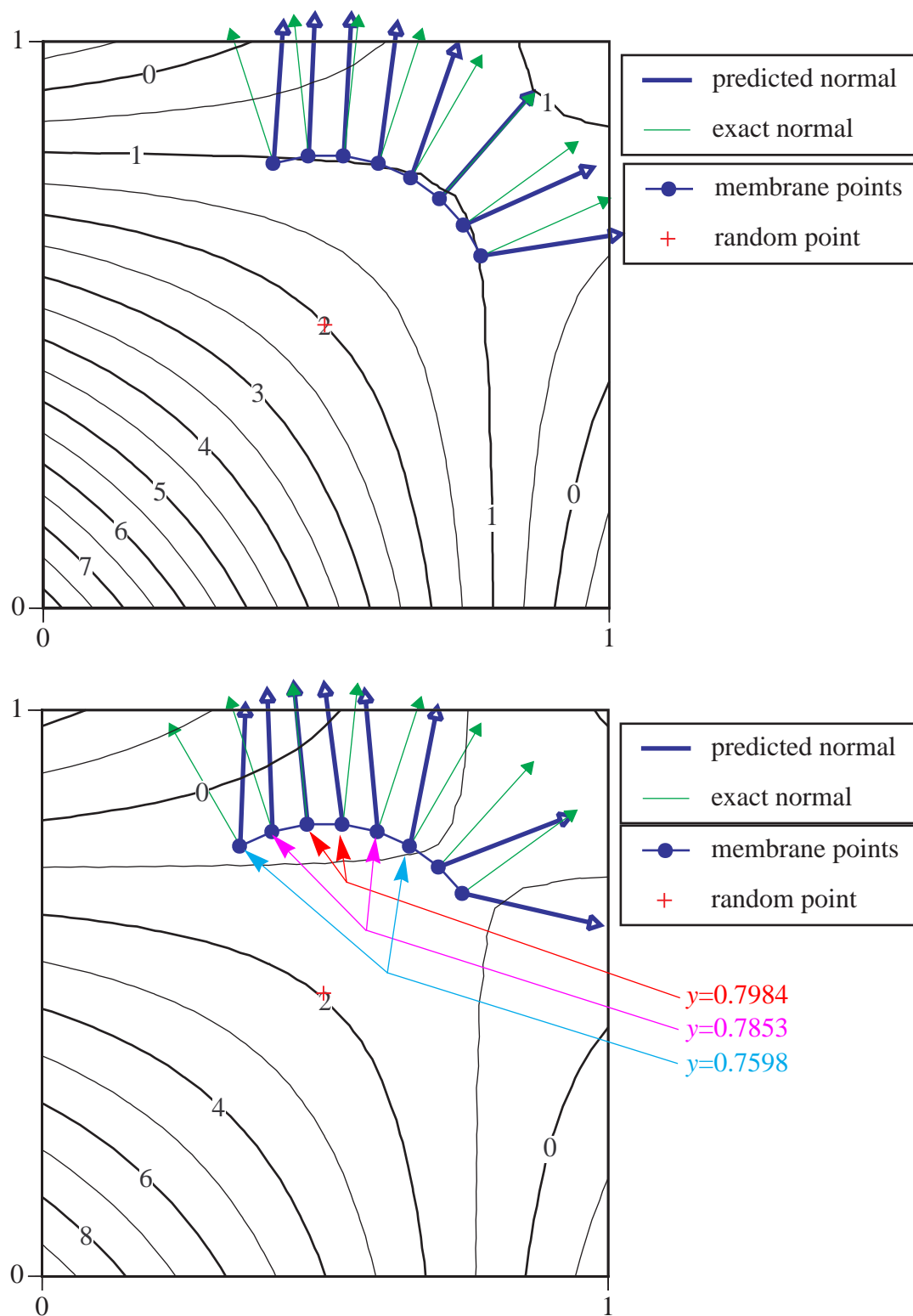


Figure 84. Normals to a Circular Arc

7.7 Using Quadratic Shape Functions in the Mass Matrix Method

Using higher order shape functions should improve the calculation of the normal vectors in certain situations when the method with linear shape functions breaks down. Equations 7.13 and 7.14 can be rewritten with higher order shape functions as

$$\chi_p^0 = \sum_c \chi_c S_c^{(2)}(\mathbf{X}_p) \quad (7.18)$$

where c ranges from one to nine. The quadratic shape functions, $S_c^{(2)}(\mathbf{X}_p)$, are defined in Appendix Section A.3.2. Here, colors are defined at nine cell centers instead of four grid vertices. The particle normals are defined to be

$$\mathbf{n}_p = \sum_p \chi_c \nabla S_c^{(2)}(\mathbf{X}_p). \quad (7.19)$$

Figure 85 illustrates the normals to a circular arc and color contours using quadratic shape functions. The membrane material point positions are identical to those of Fig. 84(bottom). Cells surrounding the one with material points which were empty are not shown. The predicted normal vectors lay on top of the theoretical ones, which is an improvement from the previous method. Also, note that the contour of color within the cell is circular. This circular contour is not possible using linear shape functions.

The dog-bone membrane expansion problem discussed in Section 6.2 (page 75) was run using the mass matrix method with linear shape functions. The results showed that the membrane expanded too much indicating errors in the normal vectors. Thus, the mass matrix method with quadratic shape functions was used, and the results were better. Figure 86 compares the normals to the membrane particles at $t=0$ using the mass matrix method with quadratic shape functions and the connectivity of the membrane points. The normals are accurate with the mass matrix method. Figure 87 compares the positions of material points at various times during the simulation using the connectivity approach and the mass matrix method with quadratic shape functions to calculate the normal vectors. The left-hand (blue/red combinations of material points) side is the simulation using connectivity, and the right-hand (cyan/dark-red combinations of material points) side is the simulation using the mass matrix method.

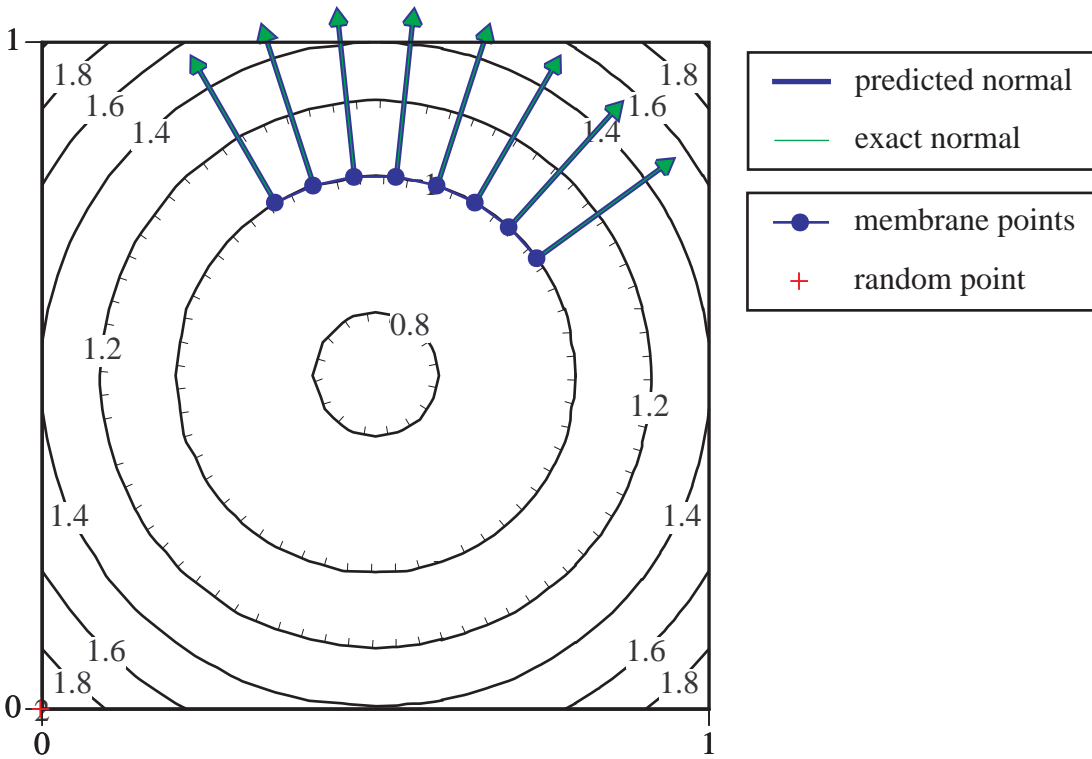


Figure 85. Normals to a Circular Arc Using Quadratic Shape Functions

In the cases when a single membrane material point was in a cell, the region considered was expanded to a group of four cells and the mass matrix method with linear shape functions was used. Only the normal for the single membrane point was calculated using the expanded cell.

Figure 87 shows that at early times the particle positions are similar, indicating that the mass matrix method is calculating normals effectively. However, at later times differences are seen in the solutions. The radius of the membrane grows larger using the mass matrix method as compared to using connectivity. The reasons for this are still under investigation.

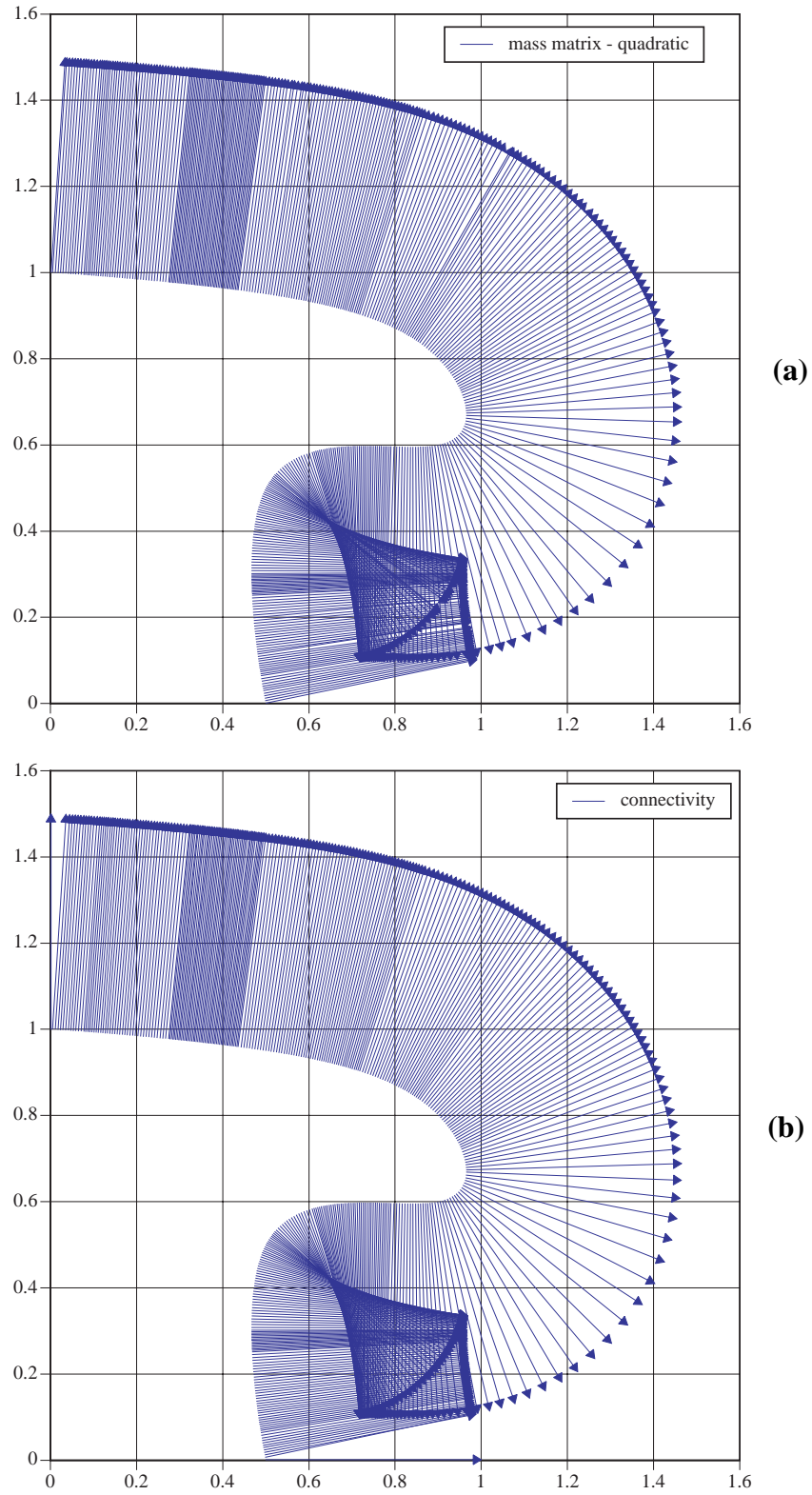


Figure 86. Normal Vectors Calculated Using: (a) Mass Matrix and (b) Connectivity

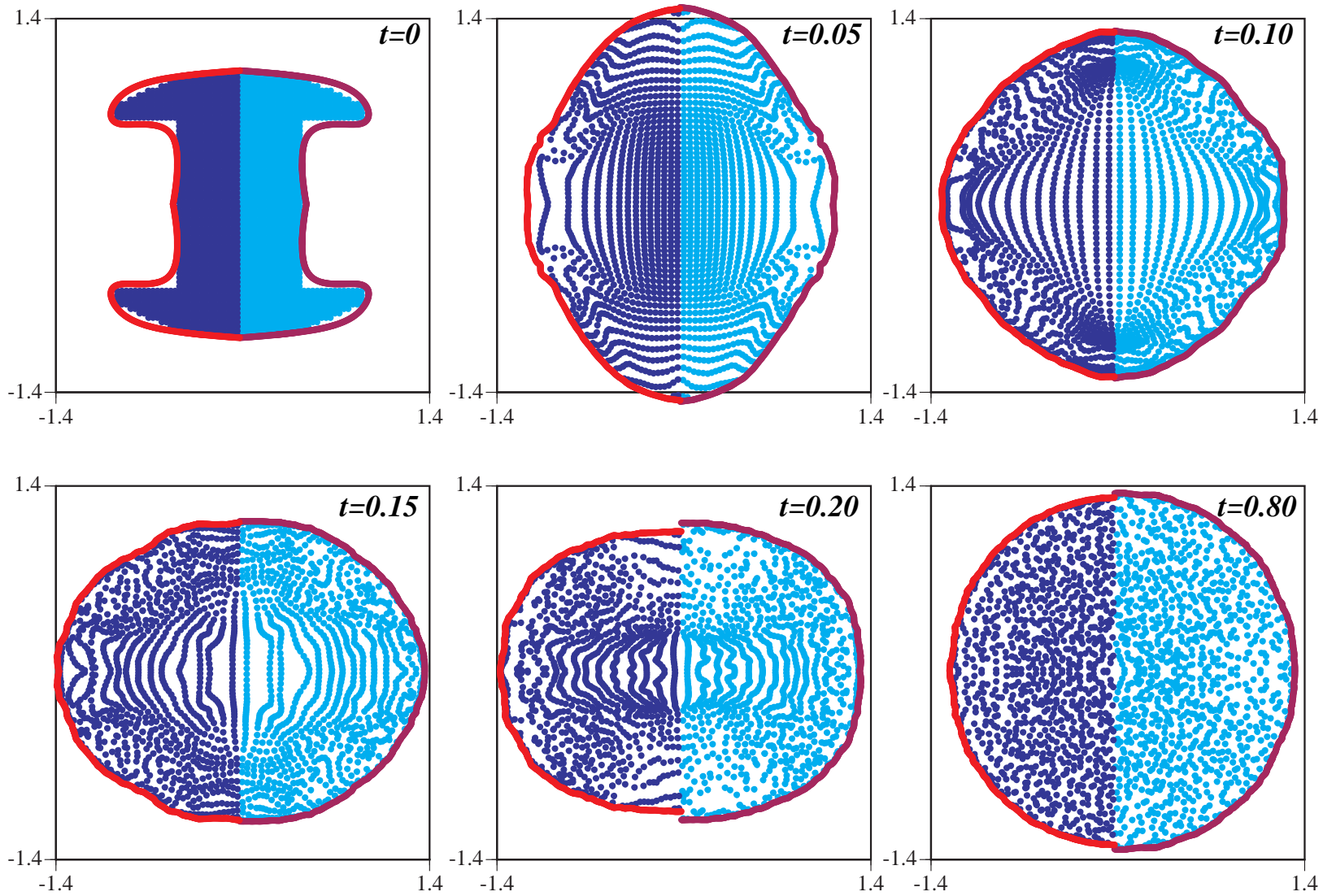


Figure 87. Membrane Expansion Simulation Using Connectivity and the Mass Matrix Method to Determine Normals

7.8 Chapter 7 Discussion and Summary

It is desired that a membrane simulation be performed without having to specify the connectivity of the material points to define the local normal-tangential coordinate system. Several methods were investigated that allow the membrane normal to be determined without using connectivity data. The quadratic and cubic methods (not using a mass matrix approach) seemed to hold promise, but they require more material points to “assist” in defining the normal. Also, the additional material points need to be clustered near the membrane to provide consistently good normal data. These methods failed in the practical application to a cylindrical membrane filled with a fluid under pressure. The only method (besides using connectivity) that worked in the practical application was the mass matrix method. When linear shape functions are used, a set of four equations must be solved simultaneously for each cell containing a membrane particle. The solution gives grid colors that when interpolated to the membrane material points result in a contour of constant color. The gradient of this color function is used to define the material particle normals. However, there are limitations to this method. For example, normals to a circular arc cannot be represented exactly.

A disadvantage to this method is when there is only one membrane material particle in a cell. The single-cell-based method has no way of defining a contour with only one particle. One solution may be to use the previously calculated normal until there are at least two membrane material points per cell. Or, a larger “ghost cell” can be defined that includes more material points. (Fig. 88).

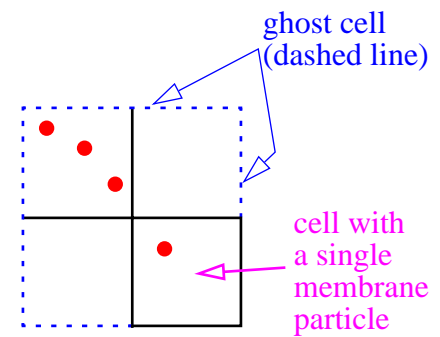


Figure 88. Ghost Cell Concept

The mass matrix method with linear shape functions is not adequate for the complex dog-bone membrane expansion problem. Therefore, a quadratic method was used with some success. The quadratic method can accurately calculate normals to a circular contour. However, there are still limitations to this method that need to be defined.

CHAPTER 8. STABILITY ANALYSIS

8.1 Background

A stability analysis of the material point method was motivated from simulations where unphysical material point accelerations were observed. After further study of the problems with these simulations it was determined that there were two causes, neither of which were stability-related. The first was with the numerical implementation of the method. In some cases division by a very small grid mass caused unnaturally large accelerations. This problem was corrected by changing to a momentum formulation where the numerator and denominator balanced. The potential benefits of the momentum formulation were known before this research started.

The second problem occurred during slow, quasi-static, or dynamic relaxation (static) simulations when a small number of material points per cell (PPC) were used (ref Section 5.4.3, page 48). Because the internal force is proportional to the number of PPC in the current implementation, unphysical internal forces which lead to noisy increments in accelerations may result (ref Section 2, page 171). These noisy accelerations cause larger relative error in slow or static simulations. This problem can be corrected by using a larger number of PPC or by using smoothing techniques.

There have not been any numerical indications of instabilities with the MPM when applying the CFL condition on the timestep based on the grid size. However, a stability analysis can frequently give insight into how to improve the method. Since a formal analysis of the MPM method has not been documented, an attempt is made here at such an analysis. The results of this section show that the CFL condition based on the grid size is the proper criterion to apply.

8.2 Introduction

Because one full time step in the material point method involves both a Lagrangian step, where the material points and grid vertices move, and a regrid step, a rigorous analytical stability analysis that considers arbitrary movement of material points is difficult. This is especially true if material points are allowed to cross cell boundaries in the analysis.

The von Neumann method is used here because it is usually the easiest method to apply (Fletcher 1991). Also, the problem is reduced to one dimension and limited to a linear elastic material.

In the von Neumann method, the errors distributed along grid lines at one time level are expanded as a finite Fourier series. The stability of the algorithm is determined by considering separate Fourier

components of the error. In a stable algorithm, the errors decay (or do not grow) from one time step to the next. In an unstable algorithm, the errors amplify.

For linear discrete governing equations, the corresponding error terms will satisfy the same equations. The procedure used in this analysis is as follows:

- 1) determine the governing equation at one grid node that advances the solution one time increment,
- 2) substitute for the solution variable the quantity $\epsilon^k e^{i\theta j}$ where $i = \sqrt{-1}$, j is the grid node designator, ϵ is the error, $\theta = \pi\Delta x$, and Δx is the grid spacing,
- 3) form the quantity, G , where $G = \frac{\epsilon^{k+1}}{\epsilon^k}$ is the gain or amplification factor,
- 4) determine the conditions for which $|G| \leq 1$, which should impose some condition on the time step.

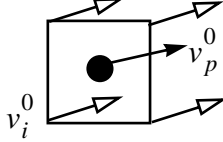
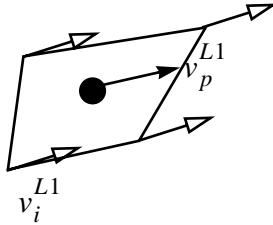
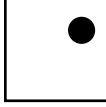
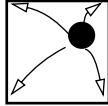
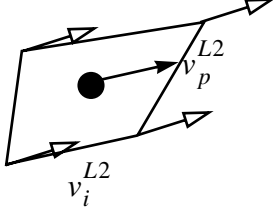
For this analysis the governing equations of the material points are used. It appears that these equations are easier to manipulate than the equations for a grid node. This is due to the fact that the velocity of a material point does not change from the Lagrangian step through the regrid step, which is not true for a grid node. This is elaborated upon in the next section.

8.3 Notation for Time Levels and Relevant Equations

Table 9 gives the material point equations and corresponding diagrams for a full time step of the MPM algorithm. It is assumed that the initial material point stress and initial internal grid forces are zero. Two-dimensional figures are used for illustration purposes only.

The material point equations at time level L2 will be written in terms of material point quantities at time level L1.

Table 9: Material Point Method - Equations for One Full Time Step

Time Level	Description	Configuration	Equations
0	Given initial material point positions, stresses, and velocities. Map initial quantities to grid.		$v_i^0 M_i^0 = \sum_p m_p v_p^0 N_i(x_p^0)$ $M_i^0 = \sum_p m_p N_i(x_p^0) \quad f_i^0 = 0$
L1	Lagrangian step: The grid equations of motion are solved. Material point velocities and positions are updated. The updated material point velocities are used to determine new grid velocities from which material point strains are calculated.		$(v_i^{L1} - v_i^0) / \Delta t = f_i^0 / M_i^0$ $v_p^{L1} = v_p^0 + \sum_i \Delta t f_i^0 N_i(x_p^{L1}) / M_i^{L1}$ $x_p^{L1} = x_p^0 + \sum_i \Delta t v_i^{L1} N_i(x_p^{L1})$ $\tilde{v}_i^{L1} = \sum_p m_p v_p^{L1} N_i(x_p^{L1}) / M_i^{L1}$ $\Delta e_p^{L1} = \Delta t \sum_j G_{jp}^{L1} \tilde{v}_i^{L1}$ $\sigma_p^{L1} = E \Delta e_p^{L1} + \sigma_p^0$
re-grid	Return grid to original or arbitrary config.		-
1	The material point quantities are mapped to the new grid vertices.		$v_p^1 = v_p^{L1}, \sigma_p^1 = \sigma_p^{L1}, x_p^1 = x_p^{L1}$ $M_i^1 = \sum_p m_p N_i(x_p^1)$ $v_i^1 = \sum_p m_p v_p^1 N_i(x_p^1) / M_i^1$ $f_i^1 = -\sum_p m_p G_{ip}^1 \sigma_p^1 / \rho_p^1$
L2	Lagrangian step: The grid equations of motion are solved. Material Point velocities and positions are updated.		$\frac{v_i^{L2} - v_i^1}{\Delta t} = \frac{f_i^1}{M_i^1}$ $v_p^{L2} = v_p^1 + \sum_i \Delta t f_i^1 N_i(x_p^1) / M_i^1$

8.4 Stability of the Material Point Equations

Consider the one dimensional problem shown in Figure 89 with grid nodes indexed $1, \dots, j-1, j, j+1, \dots, n$ and, similarly, material points designated $1, \dots, p-1, p, p+1, \dots, n-1$. Both the grid nodes and material points are equally spaced, with spacing Δx . There is one material point per cell. The following simplifying assumptions will be made at some point in the analysis (it will be clearly stated when these assumptions are made):

- The material points have the same mass, m , density, ρ , and Young's modulus, E .
- The material points initially have the same spacing relative to one another. This does not assume that they are in the center of the cell. This will lead to the assumption that the initial grid masses, M , are equal to m , and that the shape function values are initially equal. That is, N_1 for all material points is the same and N_2 for all material points is the same, but N_1 does not necessarily equal N_2 .

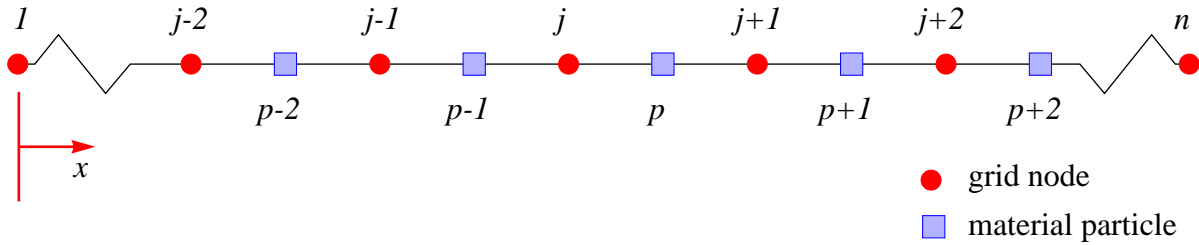


Figure 89. One-Dimensional Problem Notation

The one dimensional shape functions for element $[j, j+1]$ are

$$N_1 = N_j(x_p) = \frac{x_{j+1} - x_p}{\Delta x} \quad N_2(x_p) = 1 - N_1 = 1 - N_j \quad (8.1)$$

where x_{j+1} is the x coordinate of grid node $j+1$, and x_p is the x coordinate of material point p . These shape functions have compact support. That is, the shape functions are nonzero for material point p only in the element which contains material point p . Figure 90 illustrates the shape functions associated with each element. Note that for brevity the “(x)” has been dropped from the shape functions, and the shape functions are a function of the x coordinate. The contribution of material point p to nodes j and $j+1$ is determined by shape functions N_j and $(1-N_j)$.

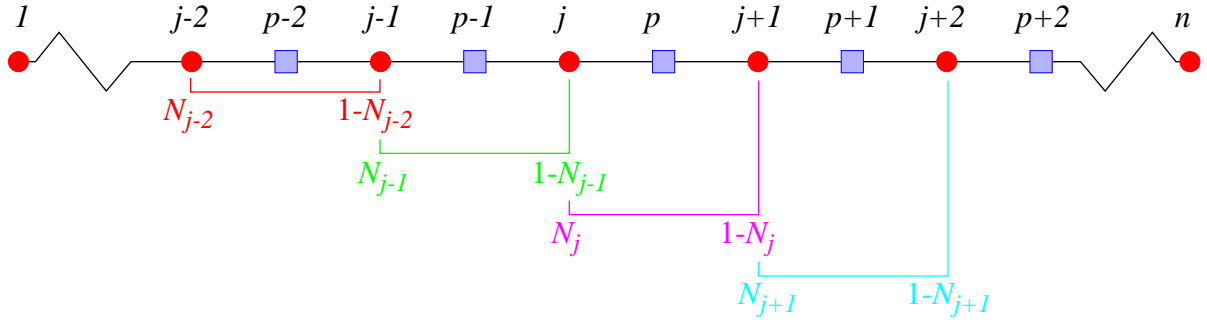


Figure 90. Shape Functions Associated with Each Element

The governing equations will be developed for material point p , in element $[j, j+1]$. First the equation for the velocity at time level L2 for material point p will be determined. Then the terms of this equation will be subsequently expanded until only terms at time level L1 are present. The velocity increment for material point p is determined by the forces at the surrounding nodes, j and $j+1$,

$$v_p^{L2} = v_p^1 + \sum_{i=j}^{j+1} \Delta t f_i^1 N_i(x_p^1) / M_i^1 \quad (8.2)$$

where superscript $L2$ designates the Lagrangian velocity at step 2, superscript 1 designates the material point velocity at step 1, Δt is the time increment and M_i^1 is the mass value at grid node i . Expanding the summation about the indicated grid nodes gives

$$v_p^{L2} = v_p^1 + \Delta t \{ f_j^1 N_j(x_p^1) / M_j^1 + f_{j+1}^1 (1 - N_j(x_p^1)) / M_{j+1}^1 \} \quad (8.3)$$

where it is understood that $N_{j+1}(x_p^1)$ is replaced by $(1 - N_j(x_p^1))$ so that the indices of the shape function and material point coordinate, j and p , are the same.

Now the forces at nodes j and $j+1$ can be written in terms of the stresses at time level 1. This gives

$$f_j^1 = - \sum_{q=p-1}^p m_q G_{jq}^1 \sigma_q^1 / \rho_q^1 \quad f_{j+1}^1 = - \sum_{q=p}^{p+1} m_q G_{j+1q}^1 \sigma_q^1 / \rho_q^1 \quad (8.4)$$

where G_{jq} is the gradient of the shape function (the one-dimensional equivalent to equation 3.14).

Making the assumptions that all material points have the same mass, m , and density, ρ , simplifies this

equation to

$$f_j^1 = -\frac{m}{\rho}(G_{j,p-1}^1\sigma_{p-1}^1 + G_{j,p}^1\sigma_p^1) \quad f_{j+1}^1 = -\frac{m}{\rho}(G_{j+1,p}^1\sigma_p^1 + G_{j+1,p+1}^1\sigma_{p+1}^1). \quad (8.5)$$

After substituting the expressions for the gradients, equation 8.5 becomes

$$f_j^1 = -\frac{m}{\rho}\left\{\left(\frac{1}{\Delta x}\right)\sigma_{p-1}^1 + \left(\frac{-1}{\Delta x}\right)\sigma_p^1\right\} \quad f_{j+1}^1 = -\frac{m}{\rho}\left\{\left(\frac{1}{\Delta x}\right)\sigma_p^1 + \left(\frac{-1}{\Delta x}\right)\sigma_{p+1}^1\right\}. \quad (8.6)$$

Updating equation 8.3 with the force values from equation 8.6 gives

$$v_p^{L2} = v_p^1 - \frac{\Delta t m}{\Delta x \rho} \left\{ \{\sigma_{p-1}^1 - \sigma_p^1\} \frac{N_j(x_p^1)}{M_j^1} + \{\sigma_p^1 - \sigma_{p+1}^1\} \frac{(1 - N_j(x_p^1))}{M_{j+1}^1} \right\}. \quad (8.7)$$

Now the stresses in equation 8.7 will be written in terms of grid velocities and subsequently material point velocities. The stress at a material point is written in terms of the surrounding grid node Lagrangian velocities, \tilde{v} . For example

$$\begin{aligned} \sigma_p^1 &= E\Delta t \sum_{i=j}^{j+1} G_{ip}^{L1} \tilde{v}_i^{L1} = E\Delta t \{G_{jp}^{L1} \tilde{v}_j^{L1} + G_{j+1,p}^{L1} \tilde{v}_{j+1}^{L1}\} = E\Delta t \left\{ -\frac{1}{\Delta x} \tilde{v}_j^{L1} + \frac{1}{\Delta x} \tilde{v}_{j+1}^{L1} \right\} \\ \sigma_p^1 &= E \frac{\Delta t}{\Delta x} \{-\tilde{v}_j^{L1} + \tilde{v}_{j+1}^{L1}\} \end{aligned} \quad (8.8)$$

where the factor of Δt takes strain rate to strain, and the factor E takes strain to stress. By shifting the indices in equation 8.8, expressions for the stresses at material points $p-1$ and $p+1$ can be written as

$$\sigma_{p-1}^1 = E \frac{\Delta t}{\Delta x} \{-\tilde{v}_{j-1}^{L1} + \tilde{v}_j^{L1}\} \quad \sigma_{p+1}^1 = E \frac{\Delta t}{\Delta x} \{-\tilde{v}_{j+1}^{L1} + \tilde{v}_{j+2}^{L1}\}. \quad (8.9)$$

Letting $c^2 = \frac{E}{\rho}$ and substituting equations 8.8 and 8.9 into 8.7 gives

$$v_p^{L2} = v_p^1 - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 m \left(\left\{ -\tilde{v}_{j-1}^{L1} + 2\tilde{v}_j^{L1} - \tilde{v}_{j+1}^{L1} \right\} \frac{N_j(x_p^1)}{M_j^1} + \left\{ -\tilde{v}_j^{L1} + 2\tilde{v}_{j+1}^{L1} - \tilde{v}_{j+2}^{L1} \right\} \frac{(1 - N_j(x_p^1))}{M_{j+1}^1} \right). \quad (8.10)$$

Now, the Lagrangian grid velocities need to be written in terms of material point velocities. For grid node $j-1$, this is written

$$\begin{aligned} \tilde{v}_{j-1}^{L1} &= \sum_{q=p-2}^{p-1} m_q v_q^{L1} N_{j-1}(x_q^{L1}) / M_{j-1}^{L1} \\ \tilde{v}_{j-1}^{L1} &= \frac{m}{M_{j-1}^{L1}} \{ v_{p-2}^{L1} (1 - N_{j-2}(x_{p-2}^{L1})) + v_{p-1}^{L1} (N_{j-1}(x_{p-1}^{L1})) \}. \end{aligned} \quad (8.11)$$

Since the shape function index always matches the material point index, we can simplify the notation by letting

$$N_{j-1}(x_{p-1}^{L1}) = N_{j-1}^{L1}. \quad (8.12)$$

Shifting indices in equation 8.11 gives the Lagrangian velocities at other grid nodes as

$$\begin{aligned} \tilde{v}_j^{L1} &= \frac{m}{M_j^{L1}} \{ v_{p-1}^{L1} (1 - N_{j-1}^{L1}) + v_p^{L1} N_j^{L1} \} \\ \tilde{v}_{j+1}^{L1} &= \frac{m}{M_{j+1}^{L1}} \{ v_p^{L1} (1 - N_j^{L1}) + v_{p+1}^{L1} N_{j+1}^{L1} \} \\ \tilde{v}_{j+2}^{L1} &= \frac{m}{M_{j+2}^{L1}} \{ v_{p+1}^{L1} (1 - N_{j+1}^{L1}) + v_{p+2}^{L1} N_{j+2}^{L1} \}. \end{aligned} \quad (8.13)$$

Equations 8.12 and 8.13 need to be substituted into equation 8.10. First, rewrite equation 8.10 as

$$v_p^{L2} = v_p^1 - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 m \left(\frac{N_j^1}{M_j^1} a^{L1} + \frac{(1 - N_j^1)}{M_{j+1}^1} b^{L1} \right) \quad (8.14)$$

where

$$a^{L1} = \left\{ -\tilde{v}_{j-1}^{L1} + 2\tilde{v}_j^{L1} - \tilde{v}_{j+1}^{L1} \right\} \quad b^{L1} = \left\{ -\tilde{v}_j^{L1} + 2\tilde{v}_{j+1}^{L1} - \tilde{v}_{j+2}^{L1} \right\}. \quad (8.15)$$

Now expand a^{L1} to give

$$a^{L1} = m \left(-\frac{1}{M_{j-1}^{L1}} \{ v_{p-2}^{L1} (1 - N_{j-2}^{L1}) + v_{p-1}^{L1} (N_{j-1}^{L1}) \} + \right. \\ \left. \frac{2}{M_j^{L1}} \{ v_{p-1}^{L1} (1 - N_j^{L1}) + v_p^{L1} N_j^{L1} \} - \right. \\ \left. \frac{1}{M_{j+1}^{L1}} \{ v_p^{L1} (1 - N_j^{L1}) + v_{p+1}^{L1} N_{j+1}^{L1} \} \right) \quad (8.16)$$

and b^{L1} to give

$$b^{L1} = m \left(-\frac{1}{M_j^{L1}} \{ v_{p-1}^{L1} (1 - N_{j-1}^{L1}) + v_p^{L1} N_j^{L1} \} + \right. \\ \left. \frac{2}{M_{j+1}^{L1}} \{ v_p^{L1} (1 - N_j^{L1}) + v_{p+1}^{L1} N_{j+1}^{L1} \} - \right. \\ \left. \frac{1}{M_{j+2}^{L1}} \{ v_{p+1}^{L1} (1 - N_{j+1}^{L1}) + v_{p+2}^{L1} N_{j+2}^{L1} \} \right). \quad (8.17)$$

Now apply the assumption that the material points are initially evenly spaced so that the grid masses equal the material point masses initially. This implies that all N_j^{L1} , $j = 1, \dots, n$ are equal, but the indices will be carried to illustrate the form of the equations. Then a^{L1} and b^{L1} simplify to

$$a^{L1} = (-\{ v_{p-2}^{L1} (1 - N_{j-2}^{L1}) + v_{p-1}^{L1} (N_{j-1}^{L1}) \} + \\ 2\{ v_{p-1}^{L1} (1 - N_j^{L1}) + v_p^{L1} N_j^{L1} \} - \\ \{ v_p^{L1} (1 - N_j^{L1}) + v_{p+1}^{L1} N_{j+1}^{L1} \}) \quad (8.18)$$

and

$$\begin{aligned}
& (-\{v_{p-1}^{L1}(1 - N_{j-1}^{L1}) + v_p^{L1}N_j^{L1}\} + \\
b^{L1} = & 2\{v_p^{L1}(1 - N_j^{L1}) + v_{p+1}^{L1}N_{j+1}^{L1}\} - \\
& \{v_{p+1}^{L1}(1 - N_{j+1}^{L1}) + v_{p+2}^{L1}N_{j+2}^{L1}\}) .
\end{aligned} \tag{8.19}$$

Let $M_j^1 = m(1 - N_j^1 + N_{j+1}^1) = m\hat{M}_j^1$ and $\frac{N_j^1}{M_j^1} = \frac{N_j^1}{m\hat{M}_j^1} = \frac{d^{(1)}}{m}$ where the superscript “(1)”

indicates this term is a function of variables at the “1” time level. Similarly, let $\frac{(1 - N_j^1)}{M_{j+1}^1} = \frac{e^{(1)}}{m}$.

Then equation 8.14 can be rewritten as

$$v_p^{L2} = v_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 (d^{(1)}a^{L1} + e^{(1)}b^{L1}) \tag{8.20}$$

where the superscript on v_p was changed from 1 to $L1$, because $v^1 = v^{L1}$ for all material points.

Next, the material point velocities in equation 8.20 need to be factored. Consider the term

$(d^{(1)}a^{L1} + e^{(1)}b^{L1})$. This is expanded to give

$$\begin{aligned}
& d^{(1)} \left(\begin{aligned} & (-\{v_{p-2}^{L1}(1 - N_{j-2}^{L1}) + v_{p-1}^{L1}(N_{j-1}^{L1})\} + \\ & 2\{v_{p-1}^{L1}(1 - N_{j-1}^{L1}) + v_p^{L1}N_j^{L1}\} - \\ & \{v_p^{L1}(1 - N_j^{L1}) + v_{p+1}^{L1}N_{j+1}^{L1}\}) \end{aligned} \right) + e^{(1)} \left(\begin{aligned} & (-\{v_{p-1}^{L1}(1 - N_{j-1}^{L1}) + v_p^{L1}N_j^{L1}\} + \\ & 2\{v_p^{L1}(1 - N_j^{L1}) + v_{p+1}^{L1}N_{j+1}^{L1}\} - \\ & \{v_{p+1}^{L1}(1 - N_{j+1}^{L1}) + v_{p+2}^{L1}N_{j+2}^{L1}\}) \end{aligned} \right)
\end{aligned} \tag{8.21}$$

which is simplified to

$$\begin{aligned}
& v_{p-2}^{L1}(-d^{(1)}(1 - N_{j-2}^{L1})) + \\
& v_{p-1}^{L1}(d^{(1)}[2(1 - N_{j-1}^{L1}) - N_{j-1}^{L1}] - e^{(1)}[(1 - N_{j-1}^{L1})]) + \\
& v_p^{L1}(d^{(1)}[2N_j^{L1} - (1 - N_j^{L1})] + e^{(1)}[2(1 - N_j^{L1}) - N_j^{L1}]) + \\
& v_{p+1}^{L1}(-d^{(1)}N_{j+1}^{L1} + e^{(1)}[2N_{j+1}^{L1} - (1 - N_{j+1}^{L1})]) + \\
& v_{p+2}^{L1}(-e^{(1)}N_{j+2}^{L1}) .
\end{aligned} \tag{8.22}$$

Thus, equation 8.20 becomes

$$v_p^{L2} = v_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 \begin{pmatrix} v_{p-2}^{L1}(-d^{(1)}(1 - N_{j-2}^{L1})) + \\ v_{p-1}^{L1}(d^{(1)}[2 - 3N_{j-1}^{L1}] - e^{(1)}[(1 - N_{j-1}^{L1})]) + \\ v_p^{L1}(d^{(1)}[3N_j^{L1} - 1] + e^{(1)}[2 - 3N_j^{L1}]) + \\ v_{p+1}^{L1}(-d^{(1)}N_{j+1}^{L1} + e^{(1)}[3N_{j+1}^{L1} - 1]) + \\ v_{p+2}^{L1}(-e^{(1)}N_{j+2}^{L1}) \end{pmatrix} \quad (8.23)$$

Note that if all material point velocities are initially equal, then $e^{(1)} = 1 - d^{(1)}$ and equation 8.23 reduces to $v_p^{L2} = v_p^{L1}$. This makes sense because if the material point velocities are equal there will be no velocity gradient, and, thus, no strains or internal forces are acting on the system.

To this point we have carried node indices for the shape functions at time level $L1$. The assumption that the material points are initially equally spaced results in

$$N_j^{L1} = N \quad j = 1, \dots, n \quad (8.24)$$

where N is the initial value.

The coefficients $d^{(1)}$ and $e^{(1)}$ can be written in terms of material point velocities at time level “ $L1$.” Recall that

$$d^{(1)} = \frac{N_j^1}{\hat{M}_j^1} = \frac{N_j^1}{(1 - N_{j-1}^1 + N_j^1)} \quad (8.25)$$

and

$$e^{(1)} = \frac{1 - N_j^1}{\hat{M}_{j+1}^1} = \frac{1 - N_j^1}{(1 - N_j^1 + N_{j+1}^1)}. \quad (8.26)$$

The expression for N_j^1 depends on the old value, $N_j^0 = N_j^{L1}$, and the velocity increments v_j^{L1} . Thus,

$$N_j^1 = \frac{x_{j+1} - x_p^1}{\Delta x} = \frac{x_{j+1} - \left(x_p^0 + \sum_{i=j}^{j+1} \Delta t v_i^{L1} N_i^{L1} \right)}{\Delta x}, \quad (8.27)$$

$$N_j^1 = N_j^0 - \frac{\Delta t}{\Delta x} \sum_{i=j}^{j+1} v_i^{L1} N_i^{L1}. \quad (8.28)$$

Replace in equation 8.28 the expression for grid node velocities and write them in terms of material point velocities. As in equation 8.11, this gives

$$N_j^1 = N - \frac{\Delta t}{\Delta x} (N(1-N)(v_{p-1}^{L1} + v_{p+1}^{L1}) + v_p^{L1}(2N^2 - 2N + 1)) \quad (8.29)$$

and the other $N_k^1, k = j-1, j+1$ have similar forms obtained by shifting the indices. Setting all initial material point velocities to zero except v_p , substituting equation 8.29 and similar forms into equations 8.25 and 8.26 and substituting this into equation 8.23 gives¹

$$v_p^{L2} = v_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 \left(\frac{v_p \left(v_p^2 \left(\frac{\Delta t}{\Delta x} \right)^2 \{6N^4 - 12N^3 + 11N^2 - 5N\} + 2v_p \left(\frac{\Delta t}{\Delta x} \right) N - 6N^2 + 6N - v_p \left(\frac{\Delta t}{\Delta x} \right) + v_p^2 \left(\frac{\Delta t}{\Delta x} \right)^2 - 2 \right)}{\left(-1 + 3v_p \left(\frac{\Delta t}{\Delta x} \right) N(N-1) + \left(\frac{\Delta t}{\Delta x} \right) v_p \right) \left(1 + 3v_p \left(\frac{\Delta t}{\Delta x} \right) N(N-1) + \left(\frac{\Delta t}{\Delta x} \right) v_p \right)} \right) \quad (8.30)$$

Now assumptions can be made on the initial location of the material points which defines the shape function values, N . For the first case consider $N=0$ which gives

$$v_p^{L2} = v_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 \left(v_p^{L1} \frac{(v_p^{L1} - 2\Delta x/\Delta t)}{(v_p^{L1} - \Delta x/\Delta t)} \right). \quad (8.31)$$

Neglecting the higher order term in equation 8.31 gives

$$v_p^{L2} = v_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 \left(\frac{(2v_p^{L1} \Delta x/\Delta t)}{(v_p^{L1} + \Delta x/\Delta t)} \right). \quad (8.32)$$

Substituting the error ϵ_p for v_p and assuming $\epsilon_p \ll 1$ gives the relation

1. Algebraic calculations performed with Maple (see Appendix A.4.14).

$$\epsilon_p^{L2} = \epsilon_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 \left(\frac{(2\epsilon_p^{L1} \Delta x / \Delta t)}{(\epsilon_p^{L1} + \Delta x / \Delta t)} \right) \approx \epsilon_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 (2\epsilon_p^{L1}) . \quad (8.33)$$

This gives an amplification factor of

$$G = 1 - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 2 . \quad (8.34)$$

For $|G| \leq 1$ two cases are evaluated. The first is

$$G \leq 1 \Rightarrow -c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 2 \leq 0 \quad (8.35)$$

which puts no condition on the time step.

The second case, $G \geq -1$, gives

$$\begin{aligned} G \geq -1 &\Rightarrow -c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 2 \geq -2 \\ c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 &\leq 1 \\ \Delta t &\leq \frac{\Delta x}{c} \end{aligned} \quad (8.36)$$

which is the Courant or CFL condition. For $N=1$, the final result is the same as in equation 8.36.

For $N=0.5$, where the material points are initially in the center of the cell, equation 8.30 reduces to

$$v_p^{L2} = v_p^{L1} - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 \left(2v_p^{L1} \frac{(v_p^{L1})^2 - 4(\Delta x / \Delta t)^2}{(v_p^{L1})^2 - 16(\Delta x / \Delta t)^2} \right) \quad (8.37)$$

which, after neglecting higher order terms, substituting ϵ_p for v_p , and assuming that $\epsilon_p \ll 1$, gives

$$\epsilon_p^{L2} \approx \epsilon_p^{L1} \left(1 - c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 2 \epsilon_p^{L1} \left(\frac{4(\Delta x / \Delta t)^2}{16(\Delta x / \Delta t)^2} \right) \right) = \epsilon_p^{L1} \left(1 - \frac{c^2}{2} \left(\frac{\Delta t}{\Delta x} \right)^2 \right) . \quad (8.38)$$

For this case the amplification factor is

$$G = 1 - \frac{c^2}{2} \left(\frac{\Delta t}{\Delta x} \right)^2 \quad (8.39)$$

which puts no condition on the time step.

$$G \leq 1 \Rightarrow -\frac{c^2}{2} \left(\frac{\Delta t}{\Delta x} \right)^2 \leq 0 . \quad (8.40)$$

The other case gives the following result.

$$\begin{aligned} G \geq -1 &\Rightarrow -\frac{c^2}{2} \left(\frac{\Delta t}{\Delta x} \right)^2 \geq -2 \\ c^2 \left(\frac{\Delta t}{\Delta x} \right)^2 &\leq 4 \\ \Delta t &\leq \frac{2\Delta x}{c} . \end{aligned} \quad (8.41)$$

This result (equation 8.41) indicates that using material points initially in the center of the cell allows for a time step twice as large as the time step that can be used when material points are initially on the sides of the cells.

8.5 Numerical Simulations Performed to Test Analytical Results

A model problem was set up test the results of the previous analysis. This elastic one-dimensional problem used five material points, all with zero initial stress. The input file is listed in the appendix on [page 189](#). The center material point was perturbed with a velocity in the $+x$ direction. Table 10 gives the parameters used in the problem.

Table 10: Stability Test Problem Parameters

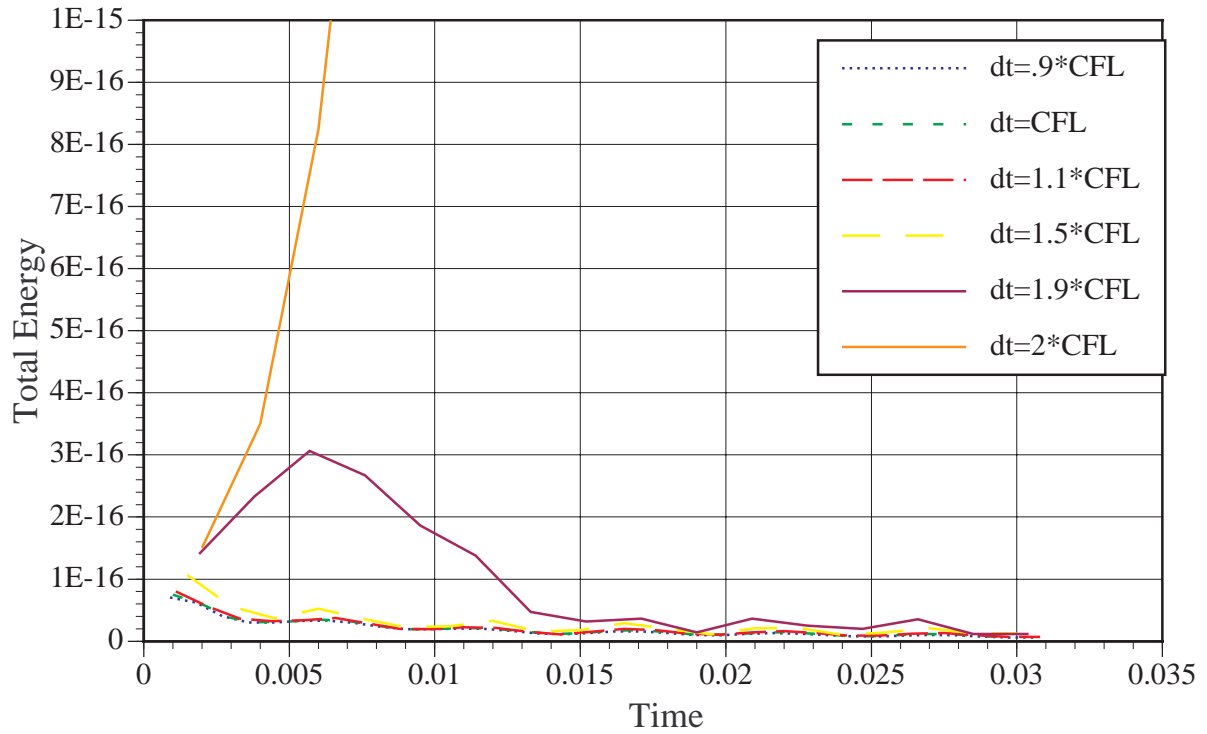
Item Description	Value
No. of Material Points	5, one per cell
E	1.0e6
v	0.0

Table 10: Stability Test Problem Parameters (Continued)

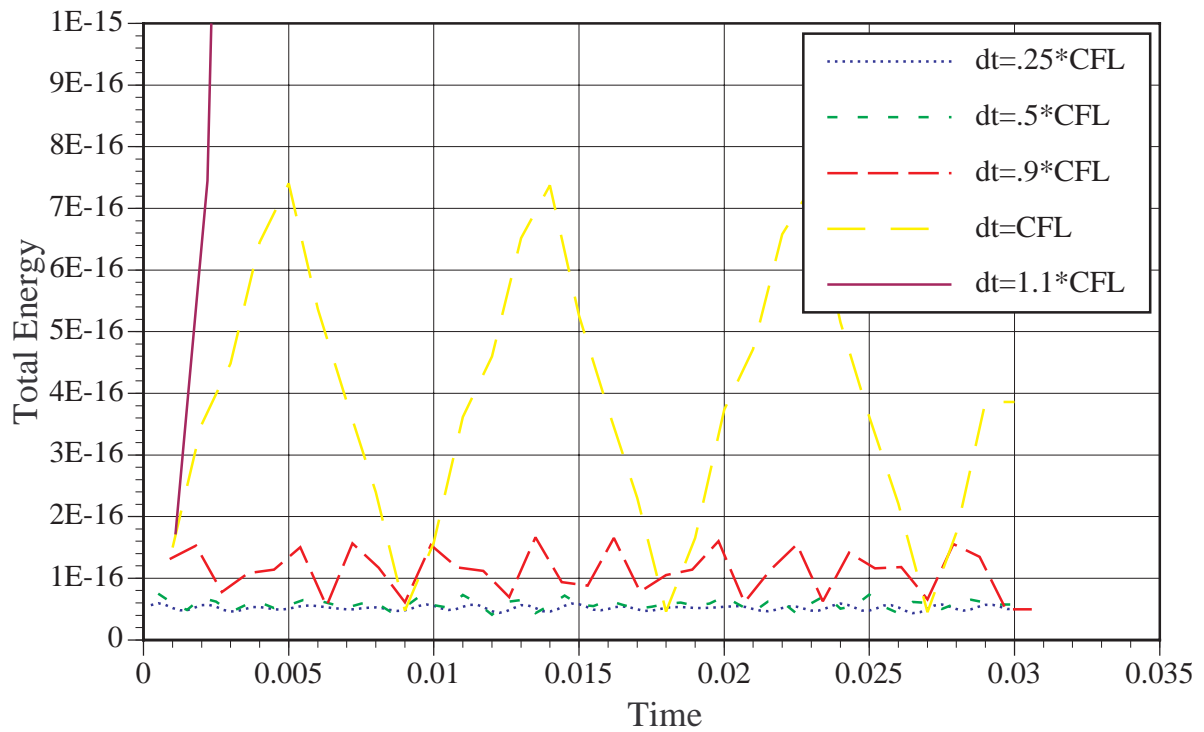
Item Description	Value
density	1.0
velocity perturbation	1.0e-8

The time step was varied and the energy was monitored over time as an indicator of stability. Figure 91 shows the energy history plots. The time step was varied from a value below that of the CFL condition to a value above it. The legends in the plots indicate the size of the time step.

For material points initially in the center of the cells, the first unstable time step observed was twice that of the CFL limit. For material points at the sides of the cells, the first unstable time step observed was a factor of 1.1 times the CFL limit. The results are close to the analytical results involving factors of 2.0 and 1.0 times the CFL limit, respectively.



(a) material points initially in center of the cells



(b) material points initially to the side of the cells

Figure 91. Numerical Results of Stability Tests for: (a) Points Initially in the Cell Centers and (b) Points Initially at the Sides of the Cells

8.6 Stability Summary

In this chapter a stability analysis of the MPM was performed with many simplifying assumptions. The analysis assumes a one-dimensional linear-elastic material with one material point perturbed from its initial position. The material point is not allowed to change cells following the perturbation. These assumptions must be made to reduce the equations to a manageable form.

The results of the analysis show that the critical time step is a function of the location of the material points. For the cases examined the smallest critical time step corresponds to the CFL condition. Numerical tests were performed simulating the particular cases that were studied analytically. Favorable agreement is seen between the critical time step determined analytically and the critical time step observed in the simulations.

CHAPTER 9. SUMMARY AND CONCLUSIONS

The first two chapters are introductory in nature. In Chapter 1 the problem of fluid-structure interaction is introduced. The material point method (MPM) is identified as a potential framework for simulating the interaction of fluids and solids. The structure considered in this work is a membrane which had not been previously modeled using the MPM. Chapter 2 summarizes the current literature and methods that are used to simulate fluid-structure interaction. One of the areas identified as needing improvement is the handling of interfaces. The most popular method (as measured by the amount of literature on the subject) for simulating the interaction of fluids and solids is a coupling between an Eulerian method for the fluid and a Lagrangian approach for the solid. In this type of coupling there is no problem with fluid mesh distortion. One disadvantage of the method is that the interface of the structure with the Eulerian mesh must be determined and the appropriate interface conditions have to be imposed. Herein lies the opportunity for improvement by using the MPM. No slip contact conditions and enforcement of the interface conditions are inherent to the MPM and are obtained at no additional computational cost.

In Chapter 3 the basic formulation for the MPM is presented and forms the baseline for the research. The formulation fits within a framework that is standard to the finite element approach. The MPM employs a fixed mesh and Lagrangian material points that move through the mesh. The weak forms of the governing equations are determined. A discrete form of the momentum equation is obtained that is solved at grid or mesh vertices, and the relationship between the grid solution and material points is presented. When a diagonal mass matrix is used, a simple explicit algorithm is the result.

In Chapter 4 the modifications necessary to simulate compressible viscous fluids are discussed. The additions to the method are a new constitutive equation and the solution of the energy equation which is necessary to employ an equation of state (i.e., the ideal gas law) for the fluid. Different methods of determining fluid density were investigated. The nearest-grid-point method is noisy, but this is improved by using the continuity equation or a higher order interpolant (e.g., quadratic) to determine material point density. The fluid formulation is validated by simulating a propagating shock (Sod's problem) and a gas expansion problem.

The modifications to the MPM necessary for the simulation of a membrane are described in Chapter 5. Prior to the beginning of the research it was unknown if the method could accurately resolve the forces along a contour of material points using a reasonable mesh size, especially if only

a single layer of points were used through the thickness. The results indicate that indeed the method is quite successful at describing a membrane. The use of multiple particles through the thickness was investigated and determined not to be feasible at this time due to the requirement of connectivity data between the material point layers. With success of a single layer of particles for representing a membrane there is also no strong need to develop a formulation with multiple layers.

A novel way to incorporate a wrinkle in a membrane is introduced. A “one-way” constitutive equation is proposed that results in a material point contributing zero force to the grid when it is in compression. The compressive strain is accumulated and released incrementally until the particle is determined to be in tension, the point at which the wrinkle is “pulled out.”

Several model problems are solved involving a membrane. A swinging pendulum problem is simulated, and convergence is demonstrated. A spring-mass system is simulated, and the results compare favorably to theory. Other simulations show results that are intuitively correct.

Chapter 6 discusses how the fluid and membrane descriptions can be combined to model fluid-membrane interaction. The approach is simple in that nothing “extra” has to be done. Fluid and membrane points are tagged as such, and the appropriate constitutive equations are invoked for each material point. The standard MPM approach to solving the governing equations using the internal forces from the fluid and membrane is used. This method for fluid-structure interaction is simple and robust. Three model problems demonstrate the utility of the method. Convergence is demonstrated in the membrane expansion problem with four consecutively refined simulations. The impact of a probe with an airbag shows reasonable agreement with experiment. The most dramatic part of these results is that a complex phenomena involving retention of gas in the airbag and impact of the probe with the airbag is simulated with a simple algorithm.

Chapter 7 summarizes the approaches that were investigated to obtain the normal to the membrane at a material point without using connectivity data. Once the normal vector is known, it is easy to ensure that the stresses in the membrane points are tangent to the membrane. Although no work has been in this research with respect to three-dimensional simulations, it was realized that having to use connectivity data of material points to define the surface in 3-D would be difficult, and it was desired not to preclude the extension of this work to 3-D.

One method was found to be superior to the others. The current implementation involves solving a four-by-four or nine-by-nine system of equations for grid or cell colors that, when interpolated to the material points, defines a contour of constant color. The gradient of this constant contour defines a

normal. This method, known as the mass matrix method, was used to simulate an oscillating fluid-filled cylindrical membrane, and the results are virtually the same as when the material point connectivity is used. The dogbone-shaped membrane expansion problem was also simulated without using connectivity.

Chapter 8 presents a stability analysis of the MPM for a linear elastic material in one dimension. Simplifying assumptions must be made to have tractable equations. The results indicate that the CFL condition governs the critical time step.

To conclude, this research has developed a version of the material point method applicable to problems involving fluid-membrane interaction. At each step of the development the results were validated. The inherent simplicity of the explicit algorithm has been retained, and the method is shown to be effective on a number of problems. However, for applications to industrial problems, a three-dimensional version of the code is necessary. This research provides the foundation that indicates the work implicit in a three-dimensional formulation is justified. Overall, the simplicity and generality of the algorithm would make solutions to complex fluid-membrane interaction problems tractable with modest computer resources.

REFERENCES

- Amsden, A.A., (1966), "The Particle-in-Cell Method for the Calculation of the Dynamics of Compressible Fluids," LA-3466, Los Alamos National Laboratory, Los Alamos, New Mexico.
- Belytschko, T., (1980), "Fluid-Structure Interaction," *Computers and Structures*, Vol 12, pp 459-469.
- Benson, D.J., (1992), "Computational Methods in Lagrangian and Eulerian Hydrocodes," *Computer Methods in Applied Mechanics and Engineering*, V 99, pp. 235-394.
- Blevins, R.D., "Formulas for natural frequency and mode shape," Robert E. Krieger Publishing, Malabar, FL.
- Brackbill, J.U., and Ruppel, H.M., (1986), "FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions," *Journal of Computational Physics*, Vol 65, pp 314-343.
- Brackbill, J.U., Kothe, D.B., Ruppel, H.M., (1988), "FLIP: A Low-Dissipation, Particle-in-Cell Method for Fluid Flow," *Computer Physics Communications*, Vol 48, pp 25-38.
- Buijk, A.J. and Low, T.C., (1993), "Significance of Gas Dynamics in Airbag Simulations," presented at the 26th ISATA, Aachen, Germany, September, 1993.
- Buijk, A.J., and Florie, C.J.L., (1991), "Inflation of Folded Driver and Passenger Airbags," Paper 91-04-B, Presented at the 1991 MSC World Users Conference, Los Angeles, California, March 1991.
- Courant, R., Friedrichs K., and Lewy, H., (1928), "Über die partiellen Differenzengleichungen der Mathematischen Physik," *Mathematische Annalen*, Vol. 100, pp. 32-74.
- Dietz, D., (1995), "Impact codes for the virtual laboratory," *Mechanical Engineering*, V 117/ No. 5, May 1995.
- de Coe, P.J.A., Nieboer, J.J., and Wismans, J., (1989), "Computer Simulation of Driver Airbag Contact with Rigid Body," TNO Report No. 751960020 to MVMA, December 1989.
- Donea, J., et al., (1977), "Lagrangian and Eulerian Finite Element Techniques for Transient Fluid-Structure Interaction Problems," *Transactions of the 4th International Congress on Structural Mechanics in Reactor Technology*, Vol B, San Francisco, CA.
- Donea, J., (1983), "Arbitrary Lagrangian-Eulerian Finite Element Methods," *Computational Methods for Transient Analysis*, eds., T. Belytschko and T.J.R. Hughes, Elsevier Science Publishers, pp 473-516.
- Fletcher, C.A.J., (1991) *Computational Techniques for Fluid Dynamics*, Vol I, Springer-Verlag, 1991.
- Florie, C.J.L., Buijk, A.J., Venis, A.C.J., and Luttwak, G., 1991, "Three-Dimensional Fluid Structure Interaction Applied in Airbag and Birdstrike Analyses," Paper 91-06, Presented at the MSC European Users Conference, Munich, Germany, June 1991.

- Forrestal, M.J., and Sagartz, M.J., (1971) "Radiated Pressure in an Acoustic Medium Produced by Pulsed Cylindrical and Spherical Shells," *Journal of Applied Mechanics*, Dec. 1971, pp. 1057-1060.
- Ghattas O., and Xiaogang, L., (1994), "A Variational Finite Element Method for Stationary Nonlinear Fluid-Solid Interaction," *Journal of Computational Physics*, Vol 121, pp. 347-356.
- Gross, M.B., and Hofmann, R., (1977) "Fluid-Structure Interaction Calculations," Transactions of the 4th International Congress on Structural Mechanics in Reactor Technology, Vol B, San Francisco, CA.
- Han, P.S., and Olson, M.D., (1987) "Interactive Analysis of Wind-Loaded Pneumatic Membrane Structures," *Computers and Structures*, Vol 25, No. 5, pp. 699-712.
- Harlow, F.H., (1964), "The Particle-in-Cell Computing Method for Fluid Dynamics," *Methods in Computational Physics*, Editors: Alder, B., Fernback, S., and Rotenberg, M., Vol 3, pp 319-343.
- Heinstein, M.W., Attaway, S.W., Swegle, J.W., and Mello, F.J., "A General-Purpose Contact Detection Algorithm for Nonlinear Structural Analysis Codes," SAND92-2141, Sandia National Laboratories, May 1993.
- Hertel, E.S., et al., (1993), "CTH: A Software Family for Multi-Dimensional Shock Physics Analysis," Proceedings 19th International Symposium on Shock Waves, Vol I, p 377, Marseille, France.
- Hirt, C.W., Amsden, A.A., and Cook, J.L., (1974), "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *Journal of Computational Physics*, V 14, pp. 227-253.
- Huerta, A., and Casadei, F., (1994), "New ALE Applications in Non-Linear Fast-Transient Solid Dynamics," *Engineering Computations*, Vol 11, p. 317.
- Huerta, A. and Liu, W.K., (1990), "Large-Amplitude Sloshing with Submerged Blocks," *Journal of Pressure Vessel Technology*, V 112, pp. 104-1-8.
- Jones, A.V., (1981), "Fluid-Structure Coupling in Lagrange-Lagrange and Euler-Lagrange Descriptions", Report EUR 7424 EN, Nuclear Science and Technology, Commission of the European Communities, Joint Research Centre, Ispra Establishment, Italy.
- Kraus, H. (1967), "Thin Elastic Shells," John Wiley and Sons.
- Landau, L.D., and Lifshitz, E.M., (1959), "Fluid Mechanics," Pergamon Press, Addison-Wesley Publishing, Reading, Mass., 1959.
- Lerch, R., (1991), "Finite Element Modelling Of Membrane Sensors Surrounded By Gaseous Or Fluid Media," *Acustica* V 73, pp. 107-113.
- Lewis, M.W., et al., (1994), "Nonlinear Dynamic Fluid-Structure Interaction Calculations with Coupled Finite Element and Finite Volume Programs," Proceedings 1994 ASME Winter Annual Meeting, Chicago, IL.
- Leveque, R., and Li, Z., (1994), "The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources," *SIAM Journal of Numerical Analysis*, Vol 31, No. 4, pp

1019-1044.

Liepmann H.W., and Roshko, A., (1957), "Elements of Gas Dynamics," John Wiley and Sons.

Liessa, A.W., (1973), "Vibration of Shells," Nasa Report NASA SP-288, Ohio State University.

Liu, W.K, and Ma, D.C., (1981), "Computer Implementation Aspects for Fluid-Structure Interaction Problems," *Computer Methods in Applied Mechanics and Engineering*, V 31, pp 129-148.

McMaster, W.H., (1984), "Computer Codes for Fluid-Structure Interactions," Proceedings 1984 Pressure Vessel and Piping Conference, San Antonio, TX, June 1984, UCRL-89724 preprint, Lawrence Livermore National Laboratory, 1984.

Neishlos, H., Israeli, M., and Kivity, Y., (1981), "Stability of Some Explicit Difference Schemes for Fluid-Structure Interaction Problems," *Computers and Structures*, Vol 13, pp 97-101.

Neishlos, H., Israeli, M., and Kivity, Y., (1983), "The Stability of Explicit Difference Schemes for Solving the Problem of Interaction Between a Compressible Fluid and an Elastic Shell," *Computer Methods in Applied Mechanics and Engineering*, Vol 41, pp 129-143.

Nieboer, J.J., et. al., (1990), "Airbag Modeling Techniques", Proceedings of the 34th Stapp Car Crash Conference, SAE Paper 902322.

Niemi, J. and Pramila, A., (1987) "FEM-Analysis of Transverse Vibrations of An Axially Moving Membrane Immersed in an Ideal Fluid," *International Journal for Numerical Methods in Engineering*, V 24, pp 2301-2313.

Nomura, T., (1994), "ALE finite element calculations of fluid-structure interaction problems," *Computer Methods in Applied Mechanics*, V 112, pp. 291-308.

Olson, L.G., and Bathe, K., (1983), "A Study of Displacement-Based Fluid Finite Elements for Calculating Frequencies of Fluid and Fluid-Structure Systems," *Nuclear Engineering and Design*, Vol 76, pp. 137-151.

Peskin, C.S., (1977), "Numerical Analysis of Blood Flow in the Heart," *Journal of Computational Mechanics*, Vol 25, pp 220-252.

Peskin, C.S., and McQueen, D.M. (1995), "A General Method for the Computer Simulation of Biological Systems Interacting with Fluids," Symposium-Society for Experimental Biology, UK, ISSN/ISBN 0081-1386, 1995.

Prasad, P. and Chou, C.C., (1989), "A Review of Mathematical Occupant Simulation Models", ASME Winter Annual Meeting, AMD-Vol 106, *Crashworthiness and Occupant Protection in Transportation Systems*, eds Khalil, T.B. and King, A.I., San Francisco, CA, 1989.

Rast, M.P., (1994), "Simultaneous Solution of the Navier-Stokes and Elastic Membrane Equations by a Finite Element Method," *International Journal for Numerical Methods in Fluids*, V 19, pp. 1115-1135.

Slavin, A.M., (1994), "Simulation of Impact of the Generic Accident-Resistant Packaging (GAP),"

SAND94-2368, Sandia National Laboratories, October 1994.

Smith, R., and Shyy, W., (1995), "Computation of unsteady laminar flow over a flexible two-dimensional membrane wing," *Phys. Fluids* **7** (9), pp. 2175-2184.

Sod, G. A., (1978), "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws," *Journal of Computational Physics*, V 27, pp. 1-31.

Sulsky, D., Chen, Z., Schreyer, H.L., (1993) "A Particle Method for History-Dependent Materials", SAND93-7044, Sandia National Laboratories, Albuquerque, New Mexico.

Sulsky, D., Chen, Z., Schreyer, H.L., (1994) "A Particle Method for History-Dependent Materials", *Computer Methods in Applied Mechanics and Engineering*, Vol 118, pp 179-186.

Taylor, L.M., and Flanagan, D.P., "PRONTO 3D A Three-Dimensional Transient Solid Dynamics Program, SAND87-1912, Sandia National Laboratories, March 1989.

Tu, C., and Peskin, C.S., (1992), "Stability and Instability in the Computation of Flows with Moving Immersed Boundaries: A Comparison of Three Methods," *SIAM Journal of Scientific and Statistical Computing*, Vol 13, No. 6, pp. 1361-1376.

Uemruea, M., (1971), "Membrane Tension And Deformation In Cylindrical Pneumatic Structures Subject To Wind Loads," IASS Pacific Symposium-Part II On Tension Structures and Space Frames, October 17-23, 1971, Tokyo and Kyoto.

van Leer, B., (1977), "Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection," *Journal of Computational Physics*, V 23, pp. 276-299.

Wilkins, M.L., (1980), "Use of Artificial Viscosity in Multidimensional Fluid Dynamic Calculations," *Journal of Computational Physics*, Vol 36, pp 281-303.

Yamamoto, K., Okada, M., and Kameyama, J., (1992), "Flow of a Viscous Fluid Past a Flexible Membrane at Low Reynolds Numbers," *Fluid Dynamics Research*, V 9, pp. 289-299.

York, A.R. and Ludwigsen, J.S., (1990), "ROAR - A 3-D Tethered Rocket Simulation Code," SAND90-1439, Sandia National Laboratories, Albuquerque, New Mexico, 87185.

Appendices

Appendix A.1 Contact-Release Algorithm	154
Appendix A.2 Example: Noise From a Particle Changing Cells	171
Appendix A.3 Interpolation Functions	173
Appendix A.4 Input Files	176

A.1 CONTACT-RELEASE ALGORITHM

A.1.1 Background

In attempting to run several example problems involving contact, it was observed that two bodies sometimes “stick” to one another unphysically when they should separate. This is caused in some situations when material points from two bodies come close to each other and are in the same computational cell.

A simple example of two impacting bars will demonstrate the problem, and then an algorithm is described that allows the bars to release from one another.

A.1.2 Two-Bar Impact With No Contact-Release Algorithm

The initial conditions for the two-bar impact problem are shown in Fig. 92. The left bar has an initial nonzero velocity v_l^0 , and the right bar is stationary. What should happen in this situation for a purely elastic impact where the bodies have no resulting strain energy is that the right bar will move to the right with velocity $v_r^1 = v_l^0$, and the left bar will have zero velocity.

What happens in reality for elastic materials is that the right bar will move off with a larger velocity than the left bar, which will be nonzero. The ratio between the initial and final velocities is the coefficient of restitution, ε , where $\varepsilon = (v_r^1 - v_l^1)/(v_l^0 - v_r^0)$.

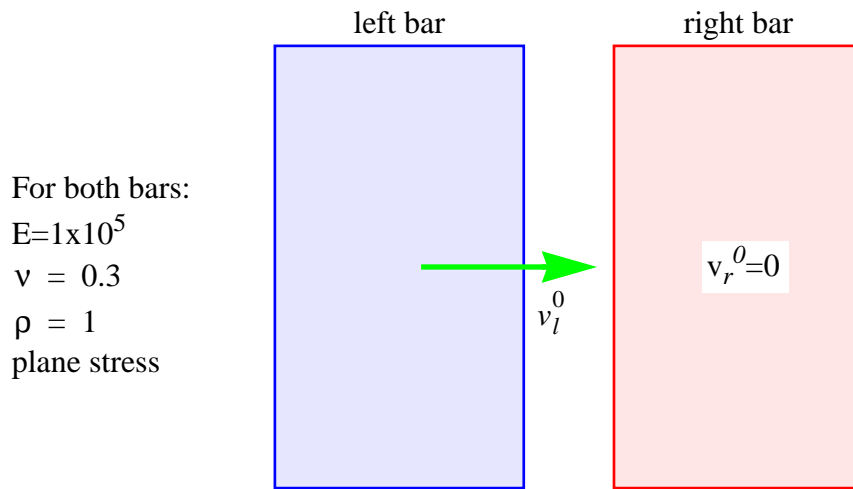


Figure 92. Initial Conditions for Two-Bar Impact

Figure 93(a) shows the initial material point configuration and the configuration at $t=0.5$. It can be seen that the two bodies do not separate. This is more evident in the lower plot of the center of mass velocity and position. The velocity of the two bodies oscillate as they move to the right together. The portion of the velocity plots between $t=0.35$ and $t=0.425$ where the two bodies have constant velocity occurred when the bodies were separated by one cell, and thus, were not in contact with each other. Contact is defined to be the case when two bodies share a grid node, i.e., when material points of two bodies both contribute to the solution at a grid node. This does not necessarily mean that material points from two bodies are in the same computational cell.

Another way to look at the problem is to consider the constitutive equation for material at the surface of a body. Free surfaces identify a state of plane stress. When two bodies come in contact, normal and shear components of the traction vector can be sustained and the constitutive equation must reflect this condition. When the bodies start to release, the normal component of traction becomes tensile. In fact the constitutive equation should be altered back to plane stress but in a general algorithm this step is never performed. The result is an “apparent” sticking which is not a physical phenomenon. Solutions consist of altering the constitutive algorithm or checking the nodal velocities. The latter approach is probably the most efficient and is described next.

A.1.3 Algorithm to Allow Release

A simple algorithm is proposed that allows bodies to release. The idea is that if the bodies are moving toward one another, as determined by some test criterion, the standard MPM method is used. That is, the material points of each body are moved in the “center-of-mass” velocity field which enforces the no-penetration condition. If the bodies are moving away from one another, they are moved in their own velocity field.

Recall from equation 3.19 that center-of-mass velocities are obtained by solving and integrating the governing equation

$$\sum_{j=1}^n m_{ij} \mathbf{a}_j = \mathbf{F}_i^{\text{int}} + \mathbf{F}_i^{\text{ext}} \quad i = 1, \dots, n \quad (9.1)$$

where $\mathbf{F}_i^{\text{int}}$ is the internal force at grid node i resulting from all material points even if they are from different bodies. New center-of-mass velocities are obtained from integrating the center-of-mass

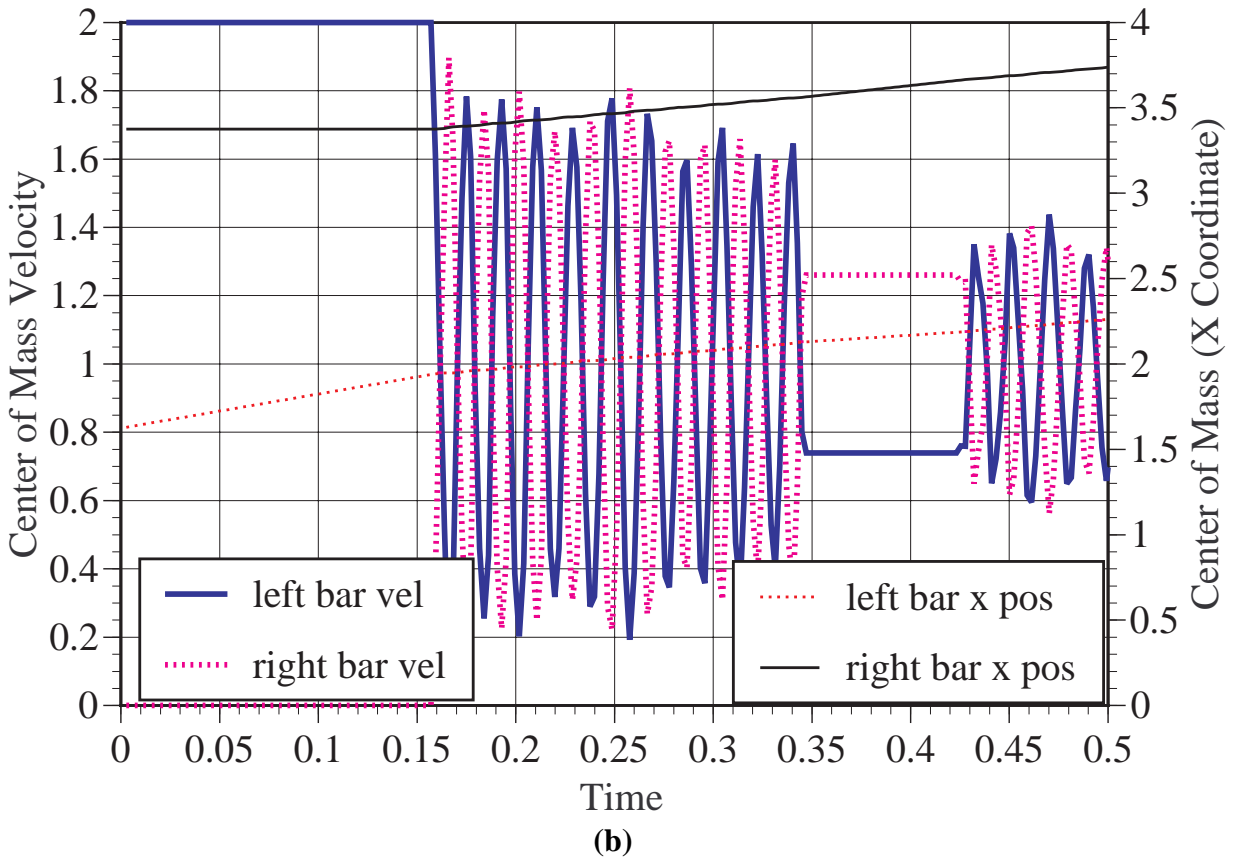
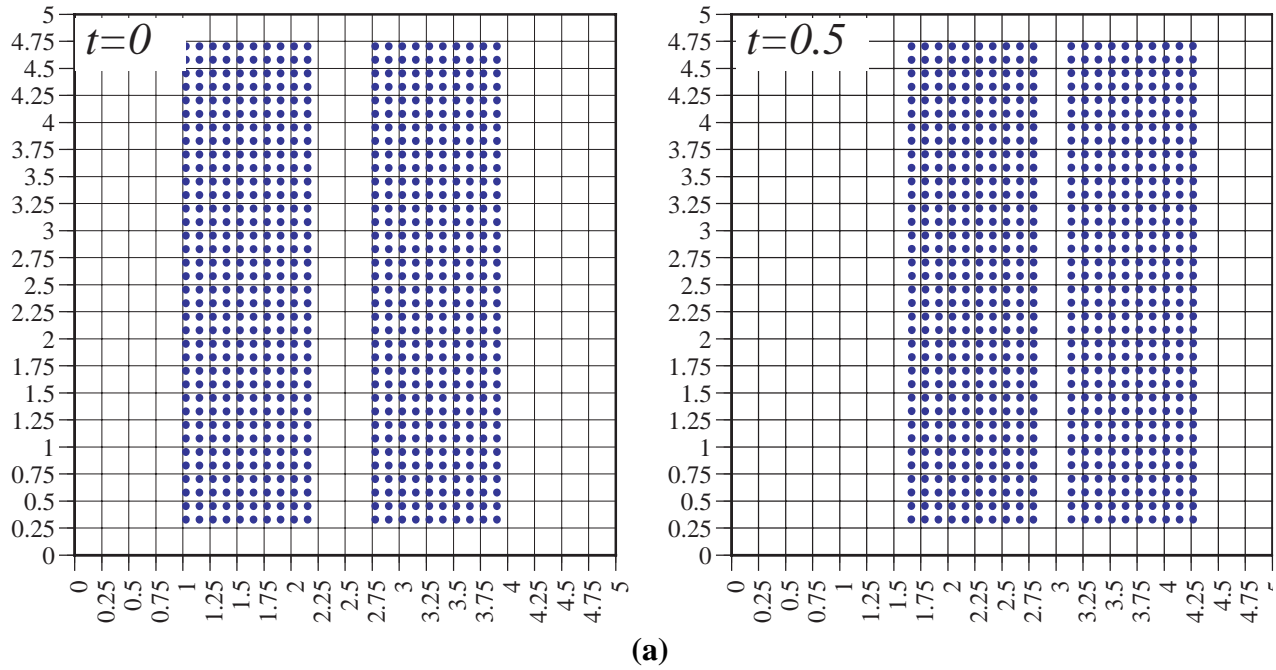


Figure 93. Two-Bar Impact Simulation with the MPM: (a) Material Point Positions, and (b) Center of Mass Velocities and Positions

accelerations

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \Delta t \mathbf{a}_i^k. \quad (9.2)$$

Recall that the velocity at time level k is determined from material point velocities at time level k as

$$\sum_{j=1}^N m_{ij}^k \mathbf{v}_j^k = \sum_{p=1}^P m_p \mathbf{v}_p^k \mathbf{N}_i(\mathbf{X}_p^k). \quad (9.3)$$

Equations 9.1 through 9.3 can also be written for a separate body I as

$$\sum_{j=1}^n m_{ij}^{(I)} \mathbf{a}_j^{(I)} = \mathbf{F}_i^{\text{int},(I)} + \mathbf{F}_i^{\text{ext}} \quad i = 1, \dots, n \quad (9.4)$$

where the internal force is from material points in body I . Equation 9.4 is integrated to give a velocity field on the grid due only to body I

$$\mathbf{v}_i^{k+1,(I)} = \mathbf{v}_i^{k,(I)} + \Delta t \mathbf{a}_i^{k,(I)} \quad (9.5)$$

where the velocity at the previous time step is determined from material points of body I

$$\sum_{j=1}^N m_{ij}^{k,(I)} \mathbf{v}_j^{k,(I)} = \sum_{p=1}^P m_p \mathbf{v}_p^{k,(I)} \mathbf{N}_i(\mathbf{X}_p^{k,(I)}). \quad (9.6)$$

If the bodies are determined to be in contact, equations 9.1 through 9.3 are used, and if they are releasing from one another equations 9.4 through 9.6 are used.

The contact-release criterion is evaluated at each grid node that is shared by two bodies. Currently, three criteria have been evaluated. The first compares the normal component of the velocity of body 1 to that of body 2. The criterion at grid node i is

$$\begin{aligned} (\mathbf{v}_i^{(1)} - \mathbf{v}_i^{(2)}) \bullet \mathbf{n}_i^{(1)} &= > 0 && \rightarrow \text{contact} \\ &= \leq 0 && \rightarrow \text{release} \end{aligned} \quad (9.7)$$

where $\mathbf{n}_i^{(1)}$ is the outward grid normal from body 1, which in this case has been designated the master body.

The second criterion is similar, but compares the velocity of the body to the center-of-mass veloc-

ity

$$\begin{aligned} (\mathbf{v}_i^{(1)} - \mathbf{v}_i^{(cm)}) \bullet \mathbf{n}_i^{(1)} &= > 0 && \rightarrow \text{contact} \\ &= \leq 0 && \rightarrow \text{release} \end{aligned} \quad (9.8)$$

where the superscript *cm* indicates center-of-mass. This formulation is attractive because the information as to which body is contacting body 1 is not needed. Here, body 1 is considered the master body. The test criteria in equations 9.7 and 9.8 gives the same result for the two-bar impact problem and the ball and net problem that will be described in Section 1.6. For each of these problems a single shared grid node was monitored at each time step throughout the entire simulation. At this grid node both criteria were evaluated and were found to give identical results as to whether or not to enforce contact or allow release. Also the time history data for position and velocity of both bodies are the same for both criteria.

The third criterion does not use the master body concept. Here, if two bodies are in close proximity to one another, each of the bodies' velocity is compared to the center of mass velocity. Release is allowed only if both criteria agree, and subsequent changes to contact, release, etc. are allowed only then. This criterion can be stated as

$$\left. \begin{aligned} (\mathbf{v}_i^{(1)} - \mathbf{v}_i^{(cm)}) \bullet \mathbf{n}_i^{(1)} \\ (\mathbf{v}_i^{(2)} - \mathbf{v}_i^{(cm)}) \bullet \mathbf{n}_i^{(2)} \end{aligned} \right\} = \begin{aligned} > 0 &\rightarrow \text{contact} \\ \leq 0 &\rightarrow \text{release} \end{aligned} \quad (9.9)$$

A.1.4 Calculating Grid Normals

The method for calculating the grid normals in equations 9.7 and 9.8 uses the color function approach described here. Each material point of a particular body is assigned a color (number) unique to that body. Interpolation of the material point color to the background grid defines the color function. The normals to bodies are determined by taking the gradient of this color function.

Let $\chi_p^{(I)}$ be the color of a material point belonging to body *I*. To smooth the function somewhat, interpolate the material point color to cell centers using quadratic interpolation

$$\chi_c^{(I)} = \sum_p \chi_p^{(I)} S_c^{(2)}(\mathbf{x}_p) \quad (9.10)$$

where each material point will contribute to nine cell centers with the use of the weighting function $S_c^{(2)}(\mathbf{X}_p)$, and $\chi_c^{(I)}$ is the cell-centered color function for body I . The normal for grid node i is calculated by mapping the cell-centered color function to grid nodes using the same gradient operator described in Section 3

$$\mathbf{n}_i^{(I)} = \sum_{c=1}^9 G_{ci} \chi_c^{(I)} \quad (9.11)$$

An example of the normals calculated using equation 9.11 is shown in Fig. 94. Due to the nine point stencil of the quadratic interpolating function more normals are calculated than would be used in the contact-release algorithm and, thus, only the pertinent normals are shown in the figure.

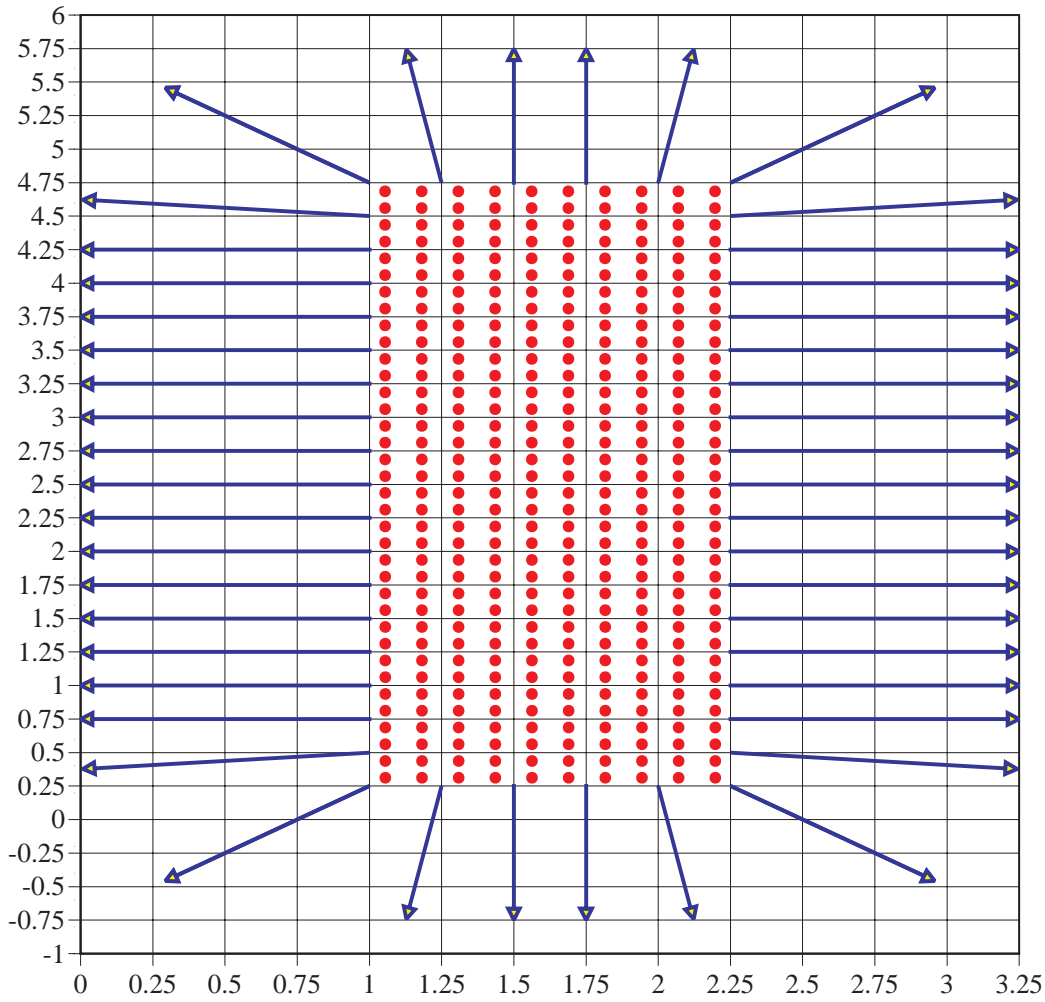


Figure 94. Grid Normals for One Bar of the Two-Bar Impact Simulation

A.1.5 Two-Bar Impact With Contact-Release Algorithm

Now let's return to the two-bar impact problem and see the results using the contact-release algorithm as described in Appendix 1.3. The criteria listed in equations 9.7 and 9.8 give the same results, which are presented here. Figure 95(a) shows the material point positions at $t=0$ and $t=0.5$. At $t=0.5$ the right bar has moved away from the left bar. This is more evident in Fig. 95(b) where the center of mass velocity of the left bar goes from 2 to about 0.3 and the right bar goes from 0 to about 1.7.

The coefficient of restitution is calculated to be $\tilde{\epsilon} = (1.71 - 0.29)/(2 - 0) = 0.71$. The tilde is applied to ϵ to indicate a pseudo coefficient of restitution that includes numerical dissipation. This value is reasonable for impact of two elastic bodies and more physical than $\tilde{\epsilon} = 0$ which is the result

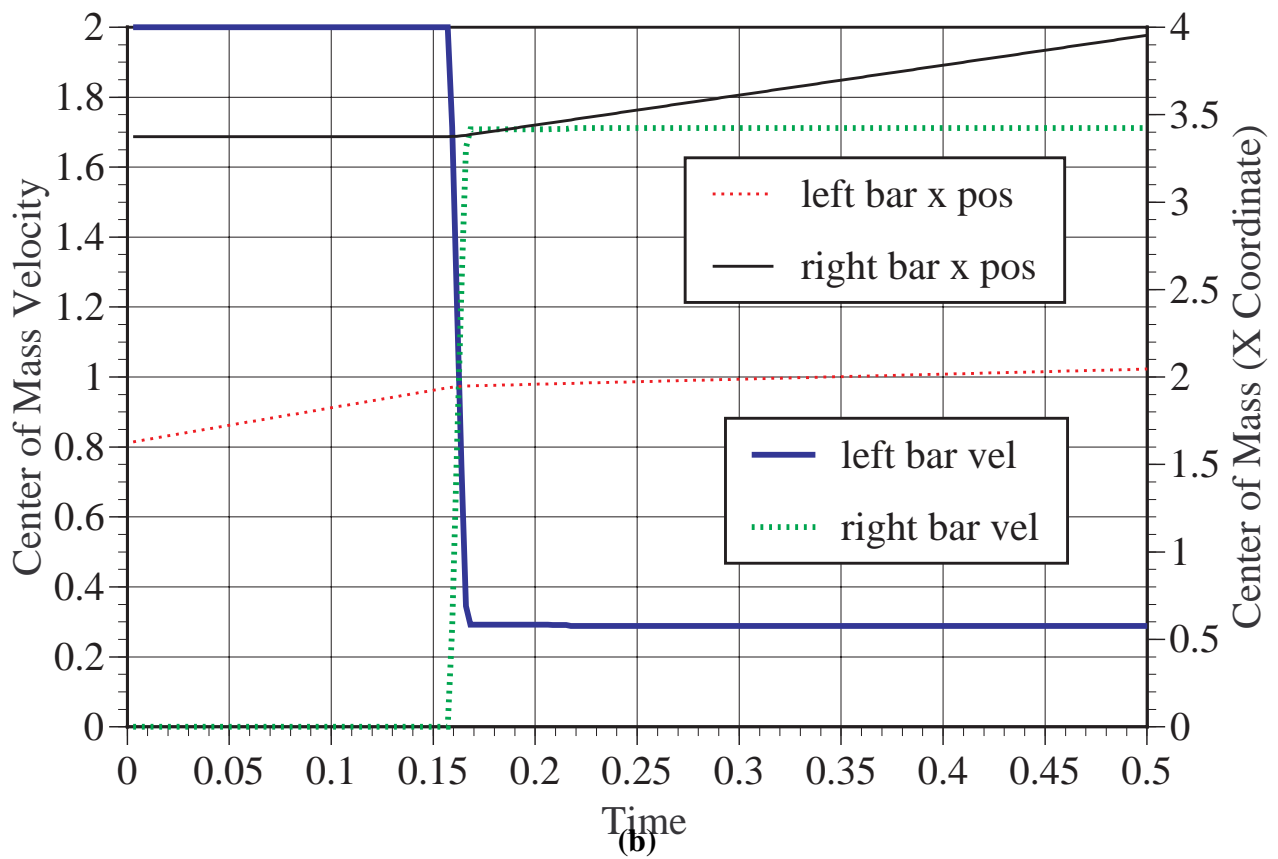
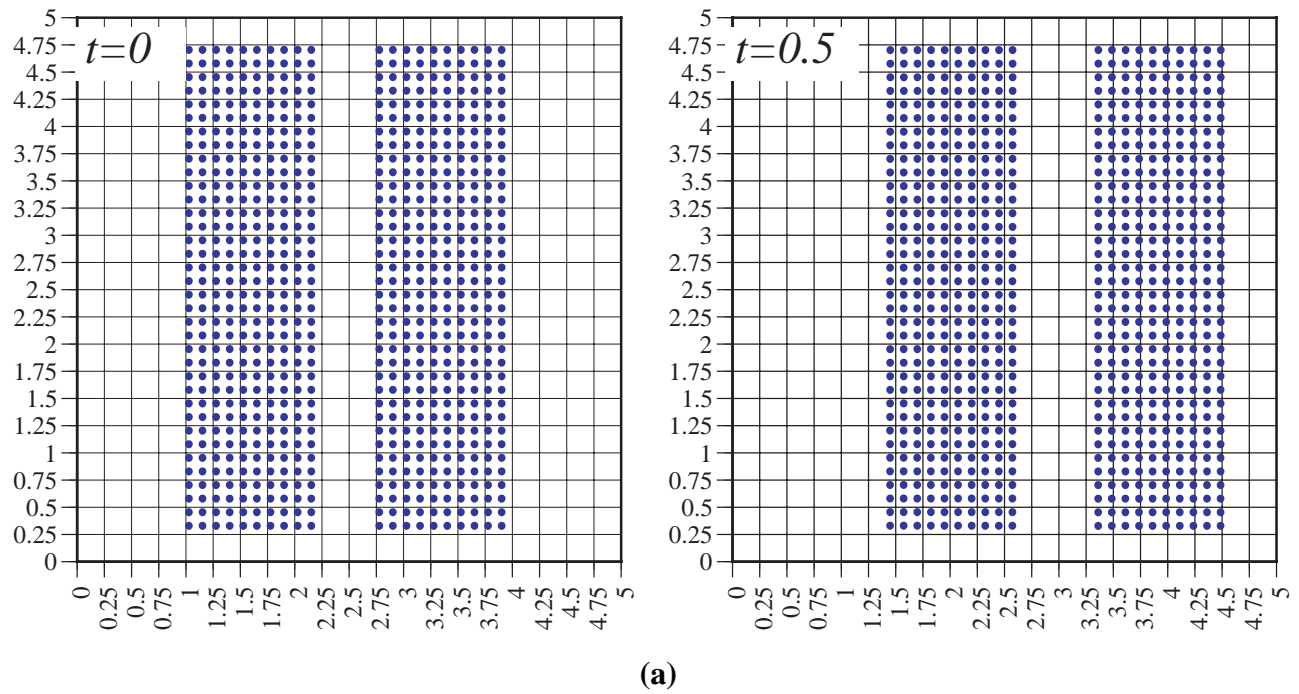
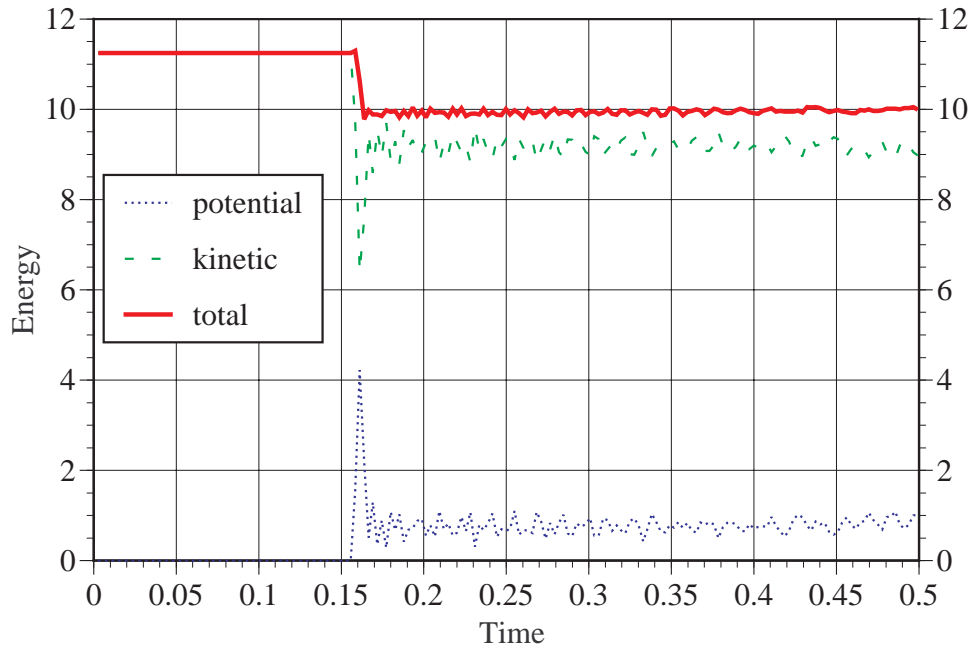


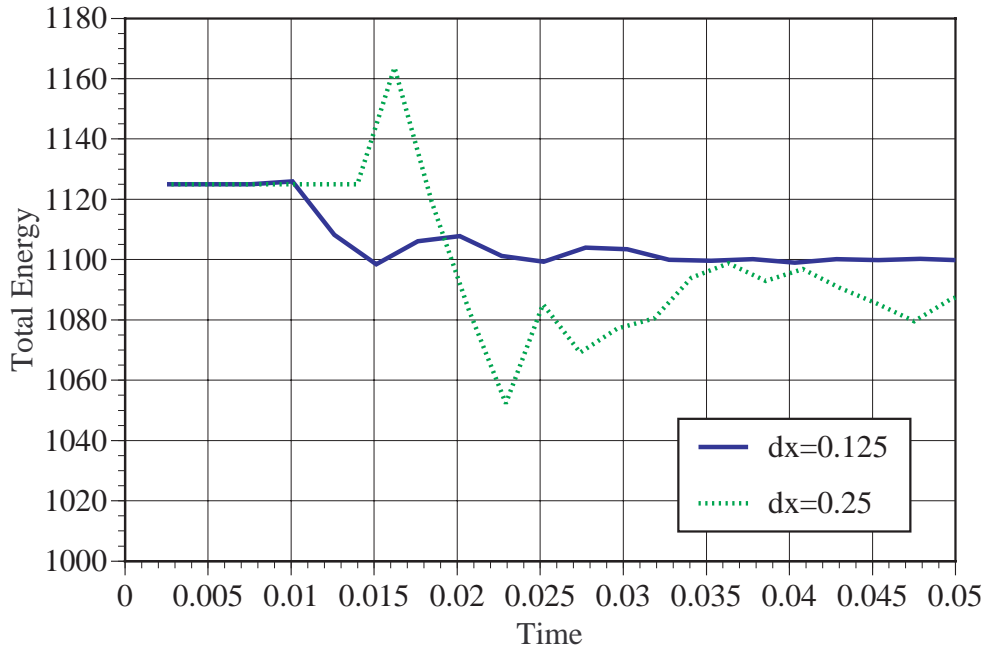
Figure 95. Two-Bar Impact Simulation with the MPM and the Contact-Release Algorithm: (a) Material Point Positions, and (b) Center of Mass Velocities and Positions

without the contact-release algorithm.

It has been observed that the contact-release algorithm is dissipative whereas for elastic bars, there is no dissipation. Figure 96(a) shows the energy history plot for the simulation above where $v_l^0 = 2$. There is about an 11% drop in energy after the impact. Figure 96(b) shows the total energy histories for two different bar impact simulations. Here $v_l^0 = 20$ and results are plotted for two mesh sizes. The dissipation appears to be less in the more refined simulation, and the amount of energy dissipated is about 2% in each case. For these simulations the more refined simulation showed $\tilde{\epsilon} = 0.81$, and the less refined showed $\tilde{\epsilon} = 0.82$. The energy spike in the less refined simulation represents only about a 3% increase and is attributed to noise.



(a)



(b)

Figure 96. (a) Energy History for Two-Bar Impact and (b) Comparison of Energy History for Simulations with $v_l^o=20$

A.1.6 Ball and Net Simulation

The physical situation for this simulation is shown in Fig. 97. A disk (called a ball) of radius $R=3$ has an initial velocity in the x direction and impacts a wire (called a net) which has a length of 10. The net is modelled as a uniaxial membrane and the ball uses a plane stress model. Both are assumed to be linear elastic.

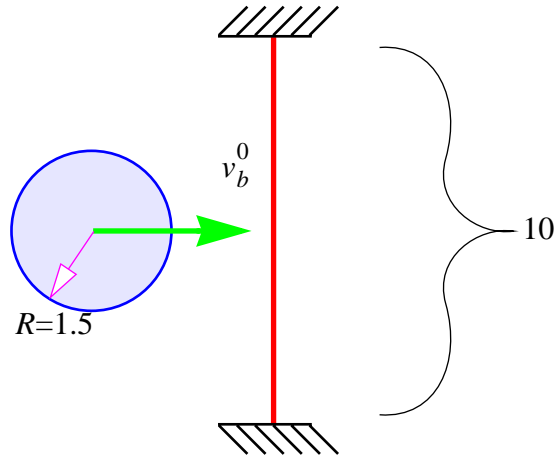


Figure 97. Ball and Net

The parameters of the problem are listed in Table 11.

Table 11: Parameters for the Ball and Net Simulation

Parameter	Ball Value	Net Value
Density	1.0	0.5
Young's Modulus	1×10^4	1×10^4
Poisson's Ratio	0.3	0.0
Initial Velocity	1.0	0.0
thickness	n/a	0.0125

It was this problem that motivated the development of a release algorithm. Originally the reason to run this simulation was to demonstrate the membrane formulation, but it was discovered that the ball would stick to the net and not release properly when rebounding to the left.

Figure 98 shows the simulation results without a contact-release algorithm. The ball actually does release from the net, but the velocity is very small (~ 0.1) as it releases to the left. The lower plot

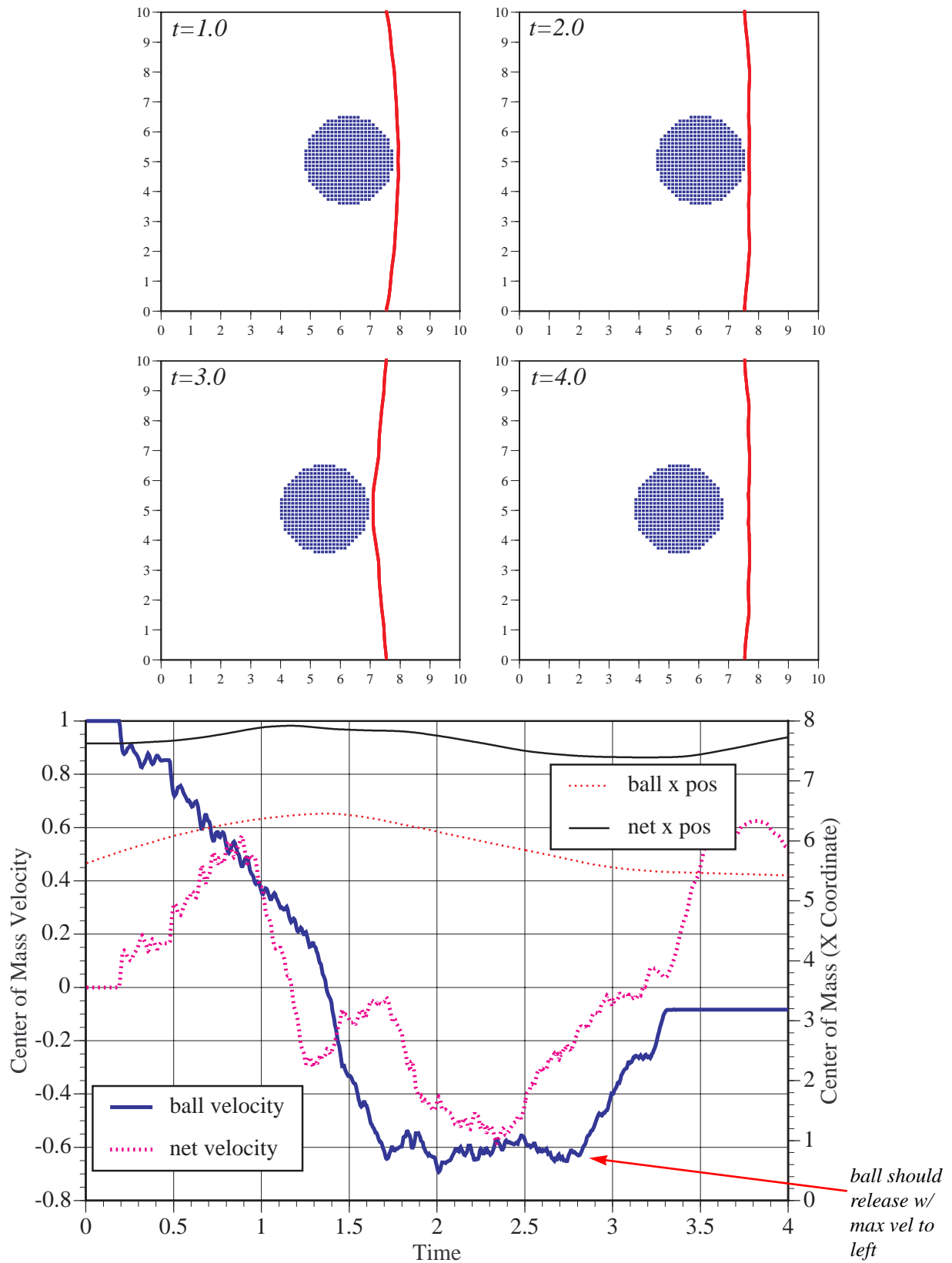


Figure 98. Ball and Net Simulation Results Without Contact-Release Algorithm

showing the ball velocity indicates that the ball does not release when the velocity is the maximum to the left, but sticks to the net which slows it down.

Figure 99 shows the simulation results with a contact-release algorithm. Figure 100(top) shows that the ball releases from the net when the ball's velocity is (approximately) maximum to the left. This is what should occur in a realistic situation. The bottom plot of this figure shows the energy history. After approximately $t=0.8$ the ball total and net total energies remain approximately constant indicating that the two bodies are not interacting.

Figure 101 shows the ball and net grid normals calculated with the algorithm described in Section 1.4. As in the previous figure showing the bar normals, the normals that would not be used in calculations are not shown. Note that the normals from the membrane are directed outward from both sides of the membrane. This gives the membrane an effective thickness for contact problems. The membrane can contact materials from both sides without the problem of having to know from which side of the membrane the normal is directed, which may occur, for example, in finite element calculations with shell elements.

In fact this method allows the membrane to be pinched between two materials which cannot be done with conventional Lagrangian finite element contact schemes. The breakdown in the finite element contact approach to pinching occurs when the thin material is forced out of one body and immediately goes inside the other body. The contact algorithm responds by forcing the thin material out of this second body and back inside the first.

The disadvantage to the contact in the MPM is that the boundaries of the material are somewhat smeared over a cell width. Thus, the fidelity of precisely defining the contact surfaces may be lost. However, the fidelity improves as the mesh is refined.

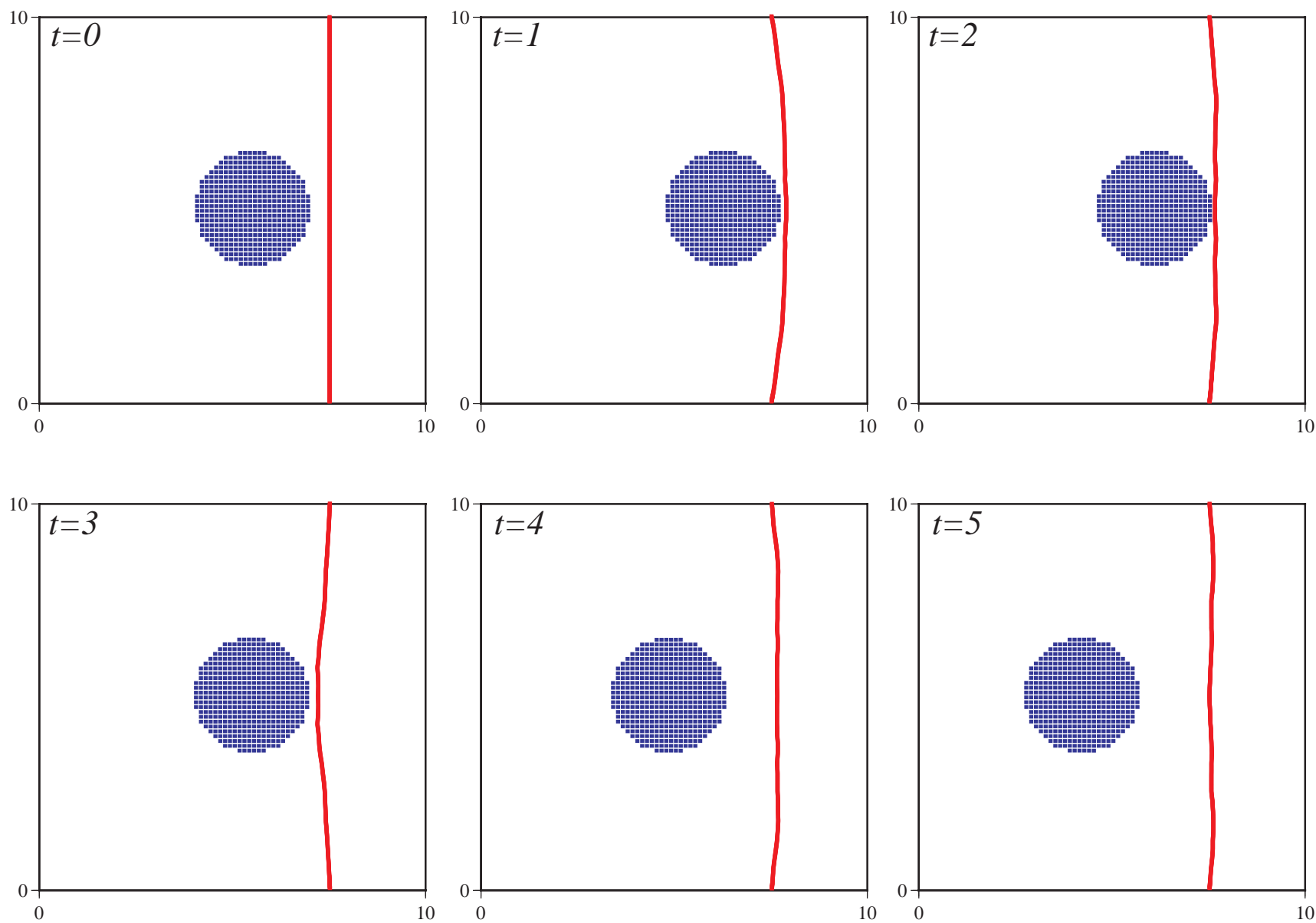


Figure 99. Ball and Net Material Point Plots With Contact-Release Algorithm

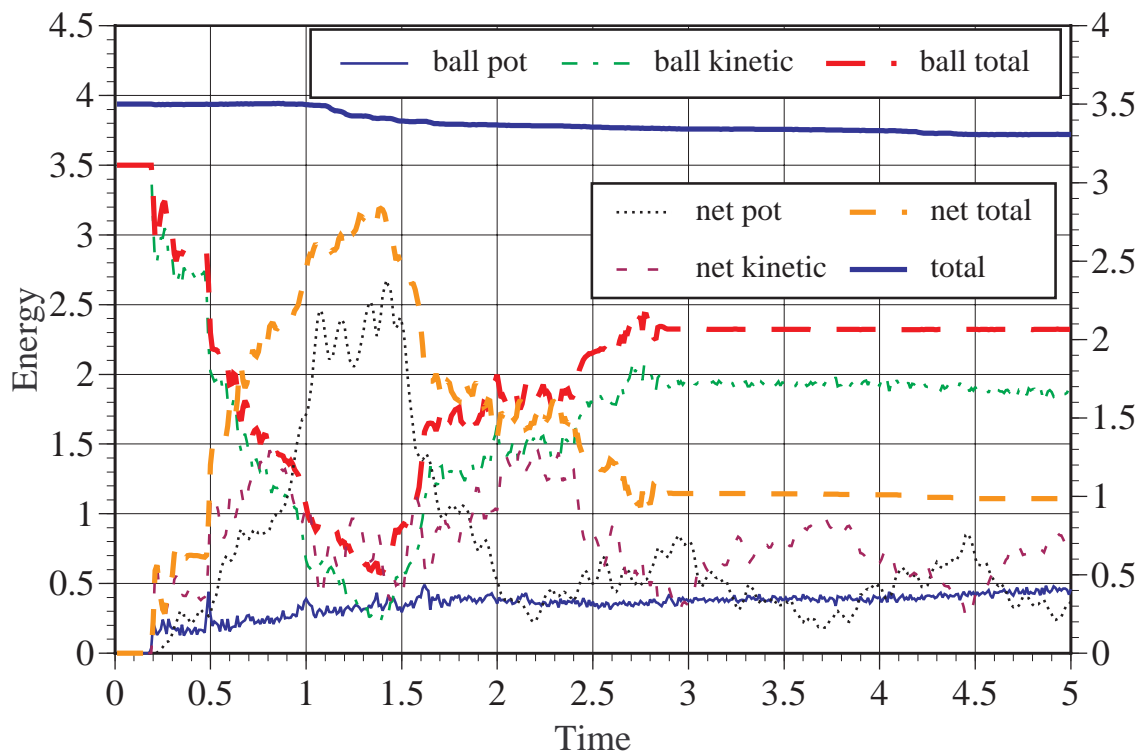
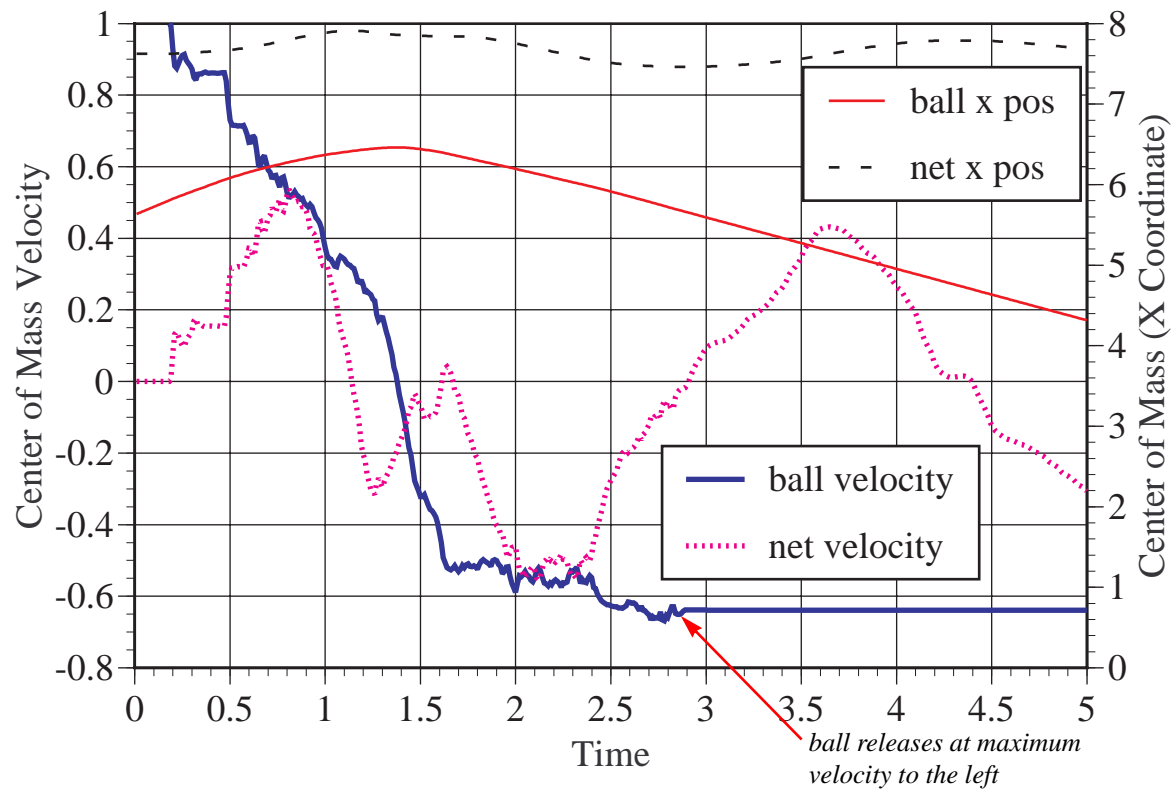


Figure 100. Ball and Net Velocity/Position and Energy Results

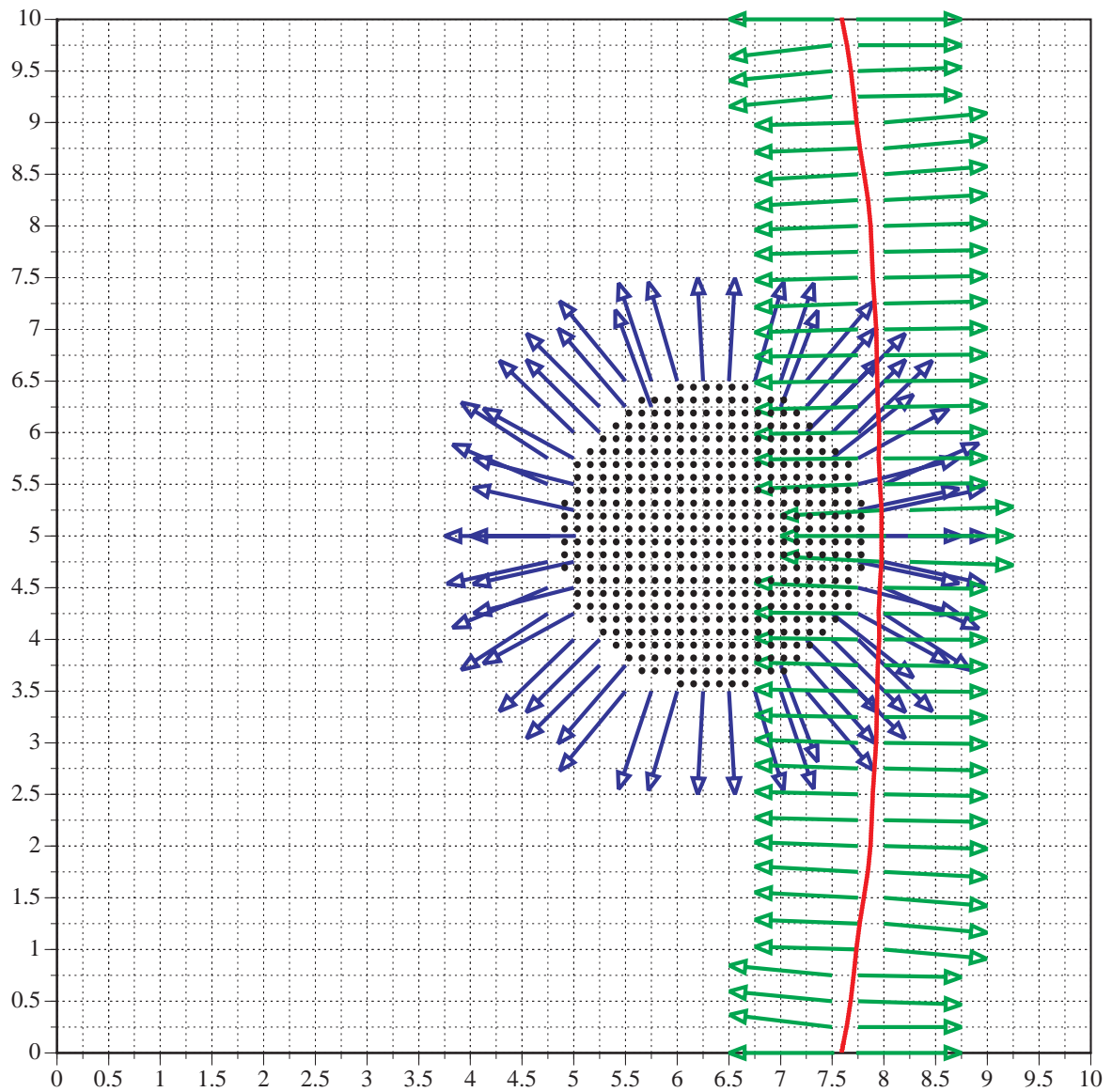


Figure 101. Ball and Net Normals at Time $t \sim 1$

A.2 EXAMPLE: NOISE FROM A PARTICLE CHANGING CELLS

This material supports the discussion in Section 5.4.3, page 48. Consider the mass-spring simulation shown in Fig. 102 that consists of ten material points numbered sequentially from top to bottom. This problem was run with the MPM code.

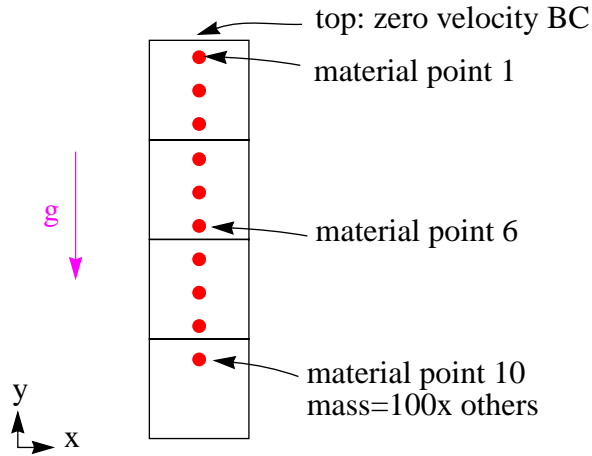


Figure 102. Set Up for a MPM Spring Simulation

At some time in the simulation, material point 6 is just about to cross cell boundaries. At this point in the simulation there are stresses, s_y , (in the y-direction) and grid node forces, f_y , as shown below.

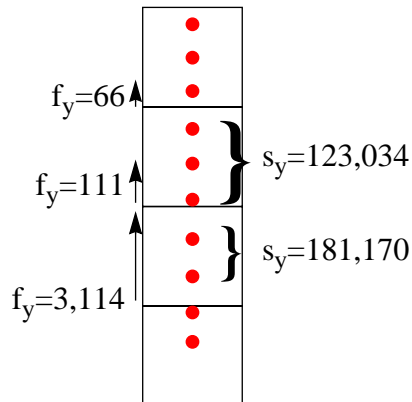


Figure 103. Forces and Stresses Prior to Material Point Crossing a Cell Boundary

The stresses are constant in each cell because the velocity gradients in the y-direction are constant in each cell.

At the next Lagrangian step the particle position and velocity are updated. Then the grid nodes are updated with new velocities to calculate strains. Next, we regrid. Now material point 6 is in the next

lower cell, but its stress was calculated based on velocity gradients from its previous cell. So the situation with stresses and new grid forces is

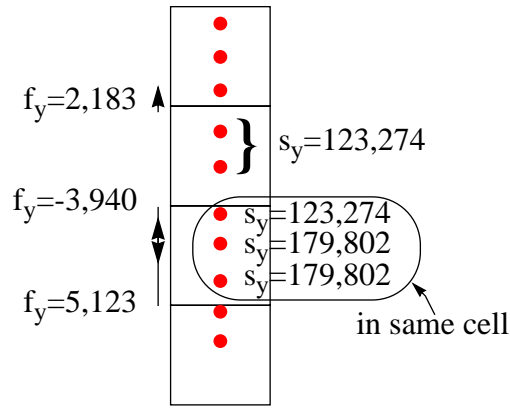


Figure 104. Forces and Stresses After Material Point 6 Crosses a Cell Boundary

Note the change in direction of forces which occurs since material point 6 is in a cell with two other material points that have a large stress (a total of 3 material points) and the cell above is left with only two particles that have a lower stress. This situation causes material point 6 to have an unphysical negative y acceleration (down), and, in fact, induces a slowly developing instability defined as an increase in energy over time.

There may be another contributing factor to the increase in energy. Just when this situation with point 6 occurs, the spring is nearly at its peak of vibration. Thus, the velocity of point 6 is small when it crosses the cell boundary. As noted in the discussion in Section 5.4.3, page 48, this problem is alleviated if force smoothing or more material points per cell are used.

A.3 INTERPOLATION FUNCTIONS

A.3.1 Linear Interpolation

For linear mapping or shape functions the convention shown in Fig. 105 is adopted. A rectangular cell of size Δx by Δy is defined with four grid nodes locally numbered from 1 to 4 in a counterclockwise manner. The cell in physical space is mapped to a unit cell in logical space. A material point p resides at logical (or local) coordinates (r,s) in the cell. The logical coordinates range from 0 to 1 with the origin at the bottom left grid node. The four linear shape functions defined over the cell in logical space are:

$$N_1 = (1-r)*(1-s),$$

$$N_2 = r*(1-s),$$

$$N_3 = r*s,$$

$$N_4 = (1-s)*r.$$

Derivatives of the shape functions are used to define a discrete gradient and divergence. The derivatives with respect to the global x direction are:

$$G_{1x} = -(1-s)/\Delta x,$$

$$G_{2x} = (1-s)/\Delta x,$$

$$G_{3x} = s/\Delta x,$$

$$G_{4x} = -s/\Delta x,$$

and the derivatives with respect to the global y direction are:

$$G_{1y} = -(1-r)/\Delta y$$

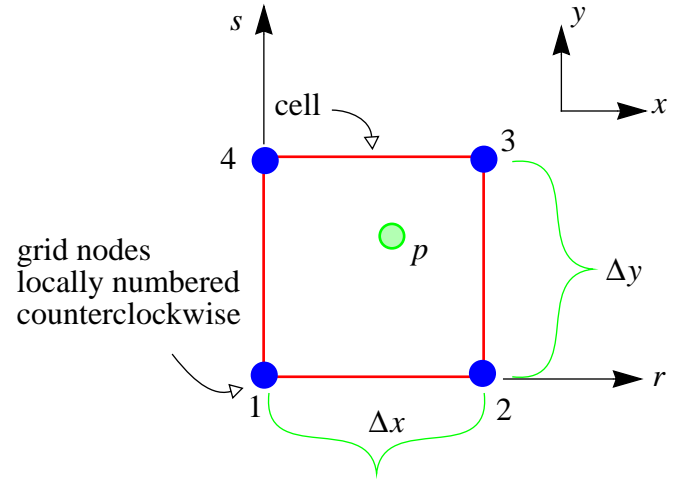


Figure 105. Nomenclature

$$G_{2y} = -r/\Delta y$$

$$G_{3y} = r/\Delta y$$

$$G_{4y} = (1-r)/\Delta y$$

A.3.2 Quadratic Interpolation

In this case the logical coordinates r and s are defined from the cell center as shown in Fig. 106(a) and range from -0.5 to +0.5. A particle in cell 1 will contribute to the cell centers of the nine neighboring cells shown in Fig. 106(b).^{*} The support of the quadratic shape function is $-3\Delta x/2 \leq r \leq 3\Delta x/2$ and similarly for s .

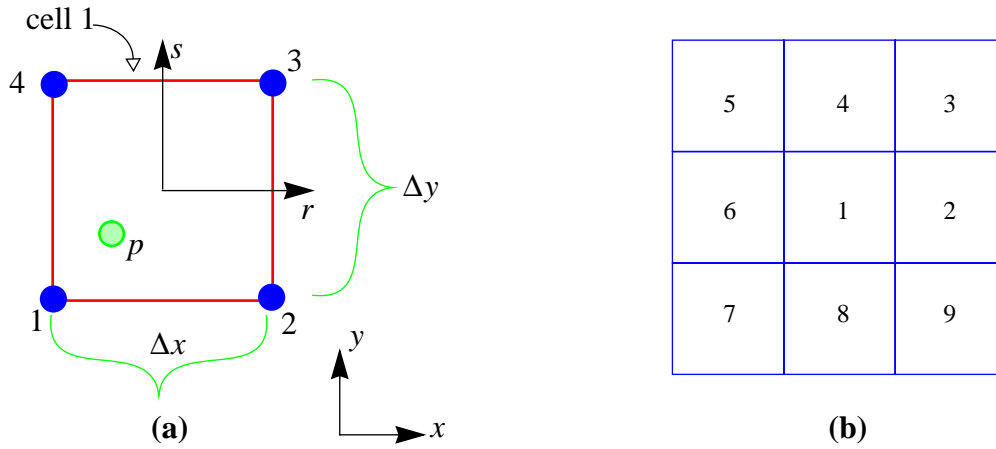


Figure 106. (a) Nomenclature and (b) Neighbor Cell Designations

The weighting functions, $S^{(2)}$, for each of the nine cell centers as a function of the material point's logical coordinates are listed in Table 12.

^{*} Note that this function is mapping to cell centers and not grid vertices.

Table 12: Quadratic Interpolation Functions for a Unit Cell in Logical Space

$S_1^{(2)} = (0.75-r^2)*(0.75-s^2)$	$S_3^{(2)} = .25*((.5+r)*(.5+s))^2$
$S_2^{(2)} = 0.5*(0.5+r)^2*(0.75-s^2)$	$S_5^{(2)} = .25*((.5-r)*(.5+s))^2$
$S_4^{(2)} = 0.5*(0.75-r^2)*(0.5+s)^2$	$S_7^{(2)} = .25*((.5-r)*(.5-s))^2$
$S_6^{(2)} = 0.5*(0.5-r)^2*(0.75-s^2)$	$S_9^{(2)} = .25*((.5+r)*(.5-s))^2$
$S_8^{(2)} = 0.5*(0.75-r^2)*(0.5-s)^2$	

These functions are denoted $S_c^{(2)}(\mathbf{X}_p)$ where the “(2)” indicates a quadratic interpolation function, and “c” indicates the cell index. Table 13 lists the derivatives of the quadratic interpolation functions.

Table 13: Derivatives Quadratic Interpolation Functions

x derivatives	y derivatives
$S_{1x}^{(2)} = -2r(.75 - s^2)/\Delta x$	$S_{1y}^{(2)} = -2s(.75 - r^2)/\Delta y$
$S_{2x}^{(2)} = (.5 + r)(.75 - s^2)/\Delta x$	$S_{2y}^{(2)} = -s(.5 + r)^2/\Delta y$
$S_{4x}^{(2)} = -r(.5 + s)^2/\Delta x$	$S_{4y}^{(2)} = (0.75 - r^2)(.5 + s)/\Delta y$
$S_{6x}^{(2)} = -(0.5 - r)(.75 - s^2)/\Delta x$	$S_{6y}^{(2)} = -s(.5 - r)^2/\Delta y$
$S_{8x}^{(2)} = -r(.5 - s)^2/\Delta x$	$S_{8y}^{(2)} = -(0.75 - r^2)(.5 - s)/\Delta y$
$S_{3x}^{(2)} = .5(.5 + r)(.5 + s)^2/\Delta x$	$S_{3y}^{(2)} = .5(.5 + r)^2(.5 + s)/\Delta y$
$S_{5x}^{(2)} = -.5(.5 - r)(.5 + s)^2/\Delta x$	$S_{5y}^{(2)} = .5(.5 - r)^2(.5 + s)/\Delta y$
$S_{7x}^{(2)} = -.5(.5 - r)(.5 - s)^2/\Delta x$	$S_{7y}^{(2)} = -.5(.5 - r)^2(.5 - s)/\Delta y$
$S_{9x}^{(2)} = .5(.5 + r)(.5 - s)^2/\Delta x$	$S_{9y}^{(2)} = -.5(.5 + r)^2(.5 - s)/\Delta y$

A.4 INPUT FILES

Input files are listed in boxes. In some cases there are two input files for each simulation. The section and the page number containing the simulation are listed in the heading.

A.4.1 Sod's Problem (Section 4.1, page 36)

```
'"ary'"
"sods problem"
$in
nx=101, ny=2, iprint=0, igen=1,gravc=0.0,
xleft=0.0e0,xright=1.0e0,ybot=0.e0,ytop=0.01,cntr=1.0,
tmax=0.143,tstep=1.d-4,dtmax=1.d-4,tplt=0.4,
tol=1.e-4,miter=30,large=.false.,wl=6,wr=6,wb=6,wt=6
iout(13)=1,tiout(13)=0.0501,
iout(30)=0,tiout(30)=0.01,
iout(33)=0,tiout(33)=0.001,
iout(34)=0,tiout(34)=0.01,
iout(35)=0,tiout(35)=5.00,
iout(36)=0,tiout(36)=0.01,
iout(37)=0,tiout(37)=2.5e-1,
iout(38)=0,tiout(38)=0.01,
iout(39)=1,tiout(39)=0.05
$
$fluid
mu=.00, gml=.4, siep(1)=2.50, siep(2)=2.0, lambda=1.00,
coef1=1.0,coef2=1.0
$
$regrig
$
$reg
nreg=2,
nvi=4,4,
xvi(1,1)=0.0e0,.5e0,0.5e0,0.0e0,
yvi(1,1)=0.0e0,0.0e0,0.01e0,0.01e0,
xvi(1,2)=0.5e0,1.0e0,1.0e0,0.50e0,
yvi(1,2)=0.0e0,0.0e0,0.01e0,0.01e0,
xcenter=0.25e0, ycenter=0.21e0,
rix(1)=0.0e0, rex(1)=200.,
riy(1)=0.0e0, rey(1)=200.,
rix(2)=0.0e0, rex(2)=200.,
riy(2)=0.0e0, rey(2)=200.,
nprg=9,9,
densir(1)=1.e0,densir(2)=0.1250,
matr(1,1)=1.e5,0.0e0,10.e25,2.e25,1.0e20,1.e2,0.e0,
matr(1,2)=1.e5,0.0e0,10.e25,2.e25,1.0e20,1.e2,0.e0,
ipath(1,1)=3,3,3,9,
ipath(1,2)=3,3,3,9,
vrax(1)= 0.0e0,
vray(1)= 0.0e0,
vrax(2)= 0.0e0,
vray(2)= 0.0e0
$
```

Box 1. Input File For Sod's Problem

A.4.2 Gas Expansion (Section 4.2, page 40)

```
'"ary"'
'"1/4 cylinder gas expansion" '
$in
nx=21, ny=21, iprint=0, igen=1,
xleft=0.500e0,xright=1.50e0,ybot=0.500e0,ytop=1.50e0,cntr=1.0,
tmax=0.02d0,tstep=0.001e0,dtmax=0.001e0,tplt=0.5,
tol=1.e-4,miter=30,large=.false.,wl=6,wr=1,wb=6,wt=1
iout(13)=1,tiout(13)=0.5,
iout(30)=0,tiout(30)=1.d-1,
iout(31)=0,tiout(31)=1.d-1,
iout(32)=0,tiout(32)=1.d-1,
iout(33)=1,tiout(33)=0.02,

$fluid
mu=.00e0, gml=.4, siep(1)=250.0, lambda=0.00,
coef1=0.0,coef2=0.0
$
$regrig
$
$reg
nreg=1,
nvi=4,
xvi(1,1)=0.0e0,1.0e0,1.0e0,0.0e0,
yvi(1,1)=0.0e0,0.0e0,1.0e0,1.0e0,
xcenter=0.5, ycenter=0.5,
rix(1)=0.0d0, rex(1)=0.376,
riy(1)=0.0d0, rey(1)=0.376,
nprg=9,
densir(1)=1.e0,
ipath(1,1)=3,3,3,7,
vrx(1)= 0.0e0,
vry(1)= 0.0e0,
$
```

Box 2. Input For Gas Expansion Simulation

A.4.3 Spring-Mass Simulation (Section 5.5, page 50)

```
'"ary"'
'"mass on a single vertical spring "'
$in
nx=2, ny=21, iprint=0, igen=0,
xleft=0.00e0,xright=0.100e0,ybot=0.00e0,ytop=2.00e0,cntr=1.0,
tmax=0.101d0,tstep=0.18d-5,dtmax=0.18d-5,tplt=0.5,
tol=1.e-4,miter=30,large=.false.,wl=1,wr=1,wb=1,wt=0
iout(13)=1,tiout(13)=0.2,
iout(30)=0,tiout(30)=0.001,
iout(31)=0,tiout(31)=1.d-1,
iout(32)=1,tiout(32)=0.2,
iout(33)=1,tiout(33)=0.0002,
iout(34)=1,tiout(34)=0.0002,
gravc=-250.0d0
methd=0
$
$fluid
$
$regrig
$
$reg
$
```

Notes on running this problem: Hardwire mass of heavy particle.
Make sure input in file 2 for no. particles per cell is correct.

Box 3. Input File 1 for Spring-Mass Problem

```
1.0d-1      !density
1.d6,0.0d0,10.d25,2.d25,1.0d25,1.d2,0.d0 !material cons
3,3,3,8     !constitutive equation
10          !no. particles in the sequence
10,3        !total no. particles, no. per cell
0.0500001.9900000.000.000.000.000.00
0.0500001.9566670.000.000.000.000.00
0.0500001.9233330.000.000.000.000.00
0.0500001.8900000.000.000.000.000.00
0.0500001.8566670.000.000.000.000.00
0.0500001.8233330.000.000.000.000.00
0.0500001.7900000.000.000.000.000.00
0.0500001.7566670.000.000.000.000.00
0.0500001.7233330.000.000.000.000.00
0.0500001.6900000.000.000.000.000.00
```

Box 4. Input File 2 for Spring-Mass Problem

A.4.4 String-Mass (Section 5.6, page 54)

The first input file is identical to that of the spring-mass simulation. The second input is different and is listed in Box 5.

```
1.0d-1    !density
1.0d6,0.00d0,10.d25,2.d25,1.0d25,1.d2,0.d0
3,3,3,8    !constitutive equation
10
10,3.0
0.050000  1.990000  0.00  0.00  0.00  0.00  0.00
0.050000  1.956667  0.00  0.00  0.00  0.00  0.00
0.050000  1.923333  0.00  0.00  0.00  0.00  0.00
0.050000  1.890000  0.00  0.00  0.00  -1.d5  0.00
0.050000  1.856667  0.00  0.00  0.00  -1.d5  0.00
0.050000  1.823333  0.00  0.00  0.00  -1.d5  0.00
0.050000  1.790000  0.00  0.00  0.00  -1.d5  0.00
0.050000  1.756667  0.00  0.00  0.00  -1.d5  0.00
0.050000  1.723333  0.00  0.00  0.00  -1.d5  0.00
0.050000  1.690000  0.00  0.00  0.00  0.00  0.00
```

Box 5. Input File 2 for String-Mass Simulation

A.4.5 Pendulum Simulation (Section 5.7, page 56)

The pendulum problem requires two input files. The first is the standard input file, and the second is file containing the initial material particle positions. Boxes 6 and 7 show the input for the pendulum simulation with 40 material particles. Input for the other simulations is obtained by changing the grid definition and time step in the first input file, and putting in the initial particle locations in the second input file.

```
'"ary"'
'"swinging pendulum" '
$in
nx=8, ny=11, iprint=0, igen=0,
xleft=0.00e0,xright=0.700e0,ybot=0.00e0,ytop=1.00e0,
tmax=1.75d0,tstep=2.e-5,dtmax=2.e-5,
tol=1.e-4,miter=30,large=.false.,wl=1,wr=1,wb=1,wt=0
iout(13)=1,tiout(13)=0.25,
iout(30)=1,tiout(30)=0.2,
iout(31)=0,tiout(31)=1.d-1,
iout(32)=1,tiout(32)=0.2,
iout(33)=1,tiout(33)=0.0100,
iout(34)=1,tiout(34)=0.0100,
iout(35)=0,tiout(35)=5.d-2,
iout(36)=0,tiout(36)=0.01,
iout(37)=0,tiout(37)=0.1,
iout(38)=1,tiout(38)=1.00d-2,
iout(40)=0,tiout(40)=0.1,
gravc=-20.0d0,methd=0
$
$fluid
$
$regrig
$
$reg
nreg=1
$

Notes on running this problem: Hardwire mass of heavy particle.
Make sure its mass is 3.3. Use np2=6.06/3.03/1.515 for 20/40/80
simulations
```

Box 6. Input File 1 for Pendulum Simulation

```

1.0d-1      !density
1.d6,0.00d0,10.d25,2.d25,1.0d25,1.d2,0.d0
3,3,3,8      !constitutive equation
40
40,3.03      !2 rows
0.333330.99990  0.0      0.0      0.0      0.0      0.0
0.327380.98203  0.0      0.0      0.0      0.0      0.0
0.321420.96416  0.0      0.0      0.0      0.0      0.0
0.315460.94629  0.0      0.0      0.0      0.0      0.0
0.309500.92842  0.0      0.0      0.0      0.0      0.0
0.303550.91055  0.0      0.0      0.0      0.0      0.0
0.297590.89268  0.0      0.0      0.0      0.0      0.0
0.291630.87482  0.0      0.0      0.0      0.0      0.0
0.285680.85695  0.0      0.0      0.0      0.0      0.0
0.279720.83908  0.0      0.0      0.0      0.0      0.0
0.273760.82121  0.0      0.0      0.0      0.0      0.0
0.267800.80334  0.0      0.0      0.0      0.0      0.0
0.261850.78547  0.0      0.0      0.0      0.0      0.0
0.255890.76760  0.0      0.0      0.0      0.0      0.0
0.249930.74973  0.0      0.0      0.0      0.0      0.0
0.243970.73186  0.0      0.0      0.0      0.0      0.0
0.238020.71399  0.0      0.0      0.0      0.0      0.0
0.232060.69612  0.0      0.0      0.0      0.0      0.0
0.226100.67825  0.0      0.0      0.0      0.0      0.0
0.220150.66038  0.0      0.0      0.0      0.0      0.0
0.214190.64252  0.0      0.0      0.0      0.0      0.0
0.208230.62465  0.0      0.0      0.0      0.0      0.0
0.202270.60678  0.0      0.0      0.0      0.0      0.0
0.196320.58891  0.0      0.0      0.0      0.0      0.0
0.190360.57104  0.0      0.0      0.0      0.0      0.0
0.184400.55317  0.0      0.0      0.0      0.0      0.0
0.178440.53530  0.0      0.0      0.0      0.0      0.0
0.172490.51743  0.0      0.0      0.0      0.0      0.0
0.166530.49956  0.0      0.0      0.0      0.0      0.0
0.160570.48169  0.0      0.0      0.0      0.0      0.0
0.154620.46382  0.0      0.0      0.0      0.0      0.0
0.148660.44595  0.0      0.0      0.0      0.0      0.0
0.142700.42808  0.0      0.0      0.0      0.0      0.0
0.136740.41022  0.0      0.0      0.0      0.0      0.0
0.130790.39235  0.0      0.0      0.0      0.0      0.0
0.124830.37448  0.0      0.0      0.0      0.0      0.0
0.118870.35661  0.0      0.0      0.0      0.0      0.0
0.112910.33874  0.0      0.0      0.0      0.0      0.0
0.106960.32087  0.0      0.0      0.0      0.0      0.0
0.101000.30300  0.0      0.0      0.0      0.0      0.0
40 mp for pendulum swing

```

Box 7. Input File 2 for Pendulum Simulation

A.4.6 Ball and Net Simulation (Section 5.9, page 67)

```
"ary"
"ball and net orig"
$in
nx=41, ny=41, iprint=0, igen=3,
xleft=0.00e0,xright=10.000e0,ybot=0.00e0,ytop=10.0e0,cntr=1.0,
tmax=4.02,tstep=0.0001d0,dtmax=0.0001d0,tplt=0.5,
tol=1.e-4,miter=30,large=.false.,wl=0,wr=1,wb=0,wt=0,
iout(13)=1,tiout(13)=1.0,
iout(31)=0,tiout(31)=1.d-1,
iout(32)=0,tiout(32)=1.d-1,
iout(33)=0,tiout(33)=0.005,
iout(34)=1,tiout(34)=0.005,
iout(37)=0,tiout(37)=2.5e-1,
iout(40)=0,tiout(40)=0.1,
methd=0,gravc=0.0
$
$fluid
$
$regrig
$
$reg
nreg=1,
nvi=4,
xvi(1,1)=0.0e0,10.0e0,10.0e0,0.0e0,
yvi(1,1)=0.0e0,0.0e0,10.0e0,10.0e0,
xcenter=5.625, ycenter=5.0,
rix(1)=0.0d0, rex(1)=1.5,
riy(1)=0.0d0, rey(1)=1.5,
nprg=4,
densir(1)=1.e0,
ipath(1,1)=3,3,3,0,
matr(1,1)=1.e4,0.30e0,10.e25,2.e25,1.0e25,1.e2,0.e0,
vrX(1)= 1.0e0
$
```

Box 8. Input File 1 for Ball and Net Simulation

A.4.7 Membrane Expansion (Section 6.2, page 75)

```
'"ary" '
' "membrane_expansion" '
$in
nx=33, ny=33 iprint=0, igen=3,
xleft=0.0d0,xright=1.60d0,ybot=0.0d0,ytop=1.60e0,cntr=1.0,
tmax=8.500014,tstep=0.35e-4,dtmax=0.35e-4,tplt=0.5,
tol=1.e-4,miter=30,large=.false.,wl=4,wr=1,wb=4,wt=1,
iout(13)=1,tiout(13)=2.00,
iout(32)=1,tiout(32)=2.00.,
iout(33)=1,tiout(33)=0.02,
iout(34)=1,tiout(34)=0.02,
iout(37)=0,tiout(37)=0.15,
iout(39)=1,tiout(39)=2.0,
iout(40)=1,tiout(40)=2.00,
iout(41)=1,tiout(41)=2.0.,
methd=0
$
$fluid
mu=0.10, gml=.4, siep(1)=0.25d3,0.25d3,
lambda=0.00,coef1=0.0,coef2=0.0
$
$regrig
$
$reg
nreg=2,
nvi=4,4,4,4,4,
xvi(1,1)=0.0, 0.426, 0.426, 0.00,
yvi(1,1)=0.0, 0.000, 0.900, 0.95,
xvi(1,2)=0.426, 0.90, 0.90, 0.426,
yvi(1,2)=0.650, 0.65, 0.74, 0.900,
xcenter=0.0,0.0,
ycenter=0.0,0.0,
rix(1)=0.0d0, rex(1)=2000.,
riy(1)=0.0d0, rey(1)=2000.,
rix(2)=0.0d0, rex(2)=2000.,
riy(2)=0.0d0, rey(2)=2000.,
nprg=64,64,
densir(1)=1.d0,densir(2)=1.d0,
ipath(1,1)=3,3,3,7,
ipath(1,2)=3,3,3,7
$
```

```
0.5
1.0d6,0.300d0,10.d25,2.d25,1.0d25,1.d2,0.d0
3,3,3,8
600
600, 6.8207
0.4999900.0011100.00.00.00.00.0
```

Box 9. Input for Membrane Expansion Simulation 4

A.4.8 Cyl-200 Airbag Simulation (Section 6.3, page 88)

```
'"ary"'
'"TNO Airbag Sim circ plate 200-restart" '
$in
nx=17, ny=26, iprint=0, igen=3,
xleft=0.00e0,xright=0.320e0,ybot=-0.001e0,ytop=0.499e0,
tmax=100.e-3,tstep=3.5e-6,dtmax=3.5e-6,
tol=1.e-4,miter=30,large=.false.,wl=5,wr=1,wb=0,wt=0,
iout(13)=1,tiout(13)=20.e-3,
iout(33)=1,tiout(33)=8.e-4,
iout(34)=1,tiout(34)=4.e-4,
iout(41)=1,tiout(41)=10.e-3,
gravc=0.0d0,method=0,cyl=1.0
$
$fluid
gml=.4,siep(1)=8226.39,siep(2)=8226.39,mu=0.0
$
$regrig
nregrig=0
$
$reg
nreg=3,
nvi=10,3,4,
xvi(1,1)=0.0 ,0.14,0.22, 0.26, 0.28 ,0.26, 0.22, 0.18,0.14, 0.0,
yvi(1,1)=0.002,0.01,0.037,0.077,0.139,0.219,0.261,0.289,0.299,0.299,
xvi(1,2)=0.08, 0.0, 0.0,
yvi(1,2)=0.299,0.318,0.299,
xvi(1,3)=0.0, 0.10,0.1,0.00,
yvi(1,3)=0.329,0.329,0.409,0.409,
xcenter=0.0,0.0,0.0, ycenter=0.0,0.0,0.0,
rix(1)=0.0, rex(1)=100.0,
riy(1)=0.0, rey(1)=100.0,
rix(2)=0.0, rex(2)=100.0,
riy(2)=0.0, rey(2)=100.0,
rix(2)=0.0, rex(3)=100.0,
riy(2)=0.0, rey(3)=100.0,
nprg=16,16,4,
densir(1)=1.2156,densir(2)=1.2156,densir(3)=2375.,
matr(1,1)=1.e0,0.3e0,0.00e0,0.00e0,1.0e20,0.e2,0.e0,
matr(1,2)=1.e0,0.3e0,0.00e0,0.00e0,1.0e20,0.e2,0.e0,
matr(1,3)=5.e7,0.3e0,10.00e25,10.00e25,1.0e20,0.e2,0.e0,
ipath(1,1)=3,3,3,7,
ipath(1,2)=3,3,3,7,
ipath(1,3)=3,3,3,0,
vry(3)=-3.90
$
```

Box 10. Input File 1 for Cyl-200 Simulation

A.4.9 Cyl-500 Airbag Simulation (Section 6.3.3, page 91)

```
'"ary" '
'"TNO Airbag Sim circ plate 500-restart" '
$in
nx=18, ny=26, iprint=0, igen=3,
xleft=0.00e0,xright=0.340e0,ybot=-0.001e0,ytop=0.499e0,
tmax=40.e-3,tstep=3.5e-6,dtmax=3.5e-6,
tol=1.e-4,miter=30,large=.false.,wl=5,wr=1,wb=0,wt=0,
iout(13)=1,tiout(13)=20.e-3,
iout(33)=1,tiout(33)=8.e-4,
iout(34)=1,tiout(34)=4.e-4,
iout(36)=0,tiout(36)=0.01,
iout(41)=1,tiout(41)=10.e-3,
gravc=0.0d0,method=0,cyl=1.0
$
$fluid
gml=.4,siep(1)=8226.39,siep(2)=8226.39,mu=0.0
$
$regrig
$
$reg
nreg=3,
nvi=10,3,4,
xvi(1,1)=0.0 ,0.14,0.22, 0.26, 0.28 ,0.26, 0.22, 0.18,0.14, 0.0,
yvi(1,1)=0.002,0.01,0.037,0.077,0.139,0.219,0.261,0.289,0.299,0.299,
xvi(1,2)=0.08, 0.0, 0.0,
yvi(1,2)=0.299,0.31,0.299,
xvi(1,3)=0.0, 0.250,0.25, 0.00,
yvi(1,3)=0.315,0.315,0.395,0.395,
xcenter=0.0,0.0,0.0, ycenter=0.0,0.0,0.0,
rix(1)=0.0, rex(1)=100.0,
riy(1)=0.0, rey(1)=100.0,
rix(2)=0.0, rex(2)=100.0,
riy(2)=0.0, rey(2)=100.0,
rix(2)=0.0, rex(3)=100.0,
riy(2)=0.0, rey(3)=100.0,
nprg=16,16,9,
densir(1)=1.2156,densir(2)=1.2156,densir(3)=360.,
matr(1,1)=1.e0,0.3e0,0.00e0,0.00e0,1.0e20,0.e2,0.e0,
matr(1,2)=1.e0,0.3e0,0.00e0,0.00e0,1.0e20,0.e2,0.e0,
matr(1,3)=5.e7,0.3e0,10.00e25,10.00e25,1.0e20,0.e2,0.e0,
ipath(1,1)=3,3,3,7,
ipath(1,2)=3,3,3,7,
ipath(1,3)=3,3,3,0,
vry(3)=-4.70
$
```

Box 11. Input File 1 for Cyl-500 Simulation

A.4.10 Piston-Container (Section 6.1, page 72)

```
'"ary"'
'"Olson-Bathe (1983) spring-piston-fluid'"
$in
nx=2, ny=41, iprint=0, igen=3,
xleft=0.00e0,xright=1.00e0,ybot=0.00e0,ytop=40.0e0,cntr=1.0,
tmax=0.035,tstep=5.d-5,dtmax=5.d-5,tplt=0.6,
tol=1.e-4,miter=30,large=.false.,wl=1,wr=1,wb=0,wt=0
iout(13)=1,tiout(13)=0.03,
iout(30)=0,tiout(30)=2.d-4,
iout(31)=0,tiout(31)=0.01,
iout(32)=0,tiout(32)=0.2,
iout(33)=1,tiout(33)=1.d-4,
iout(34)=0,tiout(34)=0.001,
iout(35)=0,tiout(35)=2.d-4,
iout(36)=0,tiout(36)=0.01,
iout(37)=0,tiout(37)=0.001,
iout(38)=0,tiout(38)=0.01,
iout(39)=0,tiout(39)=1.d-1
iout(40)=0,tiout(40)=0.1,
gravc= 0.0d0
$
$fluid
mu=0.00e0, gml=.4, siep(1)=0.10, lambda=0.00,
coef1=0.0,coef2=0.0
$
$reg
nreg=1,
nvi=4,
xvi(1,1)= 0.35d0, 0.650d0, 0.650d0, 0.350d0,
yvi(1,1)= 0.d0, 0.0d0, 20.0d0, 20.0d0,
xcenter=0.5, ycenter=5.0,
rix(1)=0.00d0, rex(1)=100.0d0,
riy(1)=0.00d0, rey(1)=100.0d0,
nprg=1,
densir(1)=1.d-4,
ipath(1,1)=3,3,3,8,
matr(1,1)=1.58e6,0.30e0,10.e25,2.e25,1.0e25,1.e2,0.e0
vrx(1)= 0.0e0,
vry(1)= 0.0e0,
vrx(2)= 0.0e0,
vry(2)= 0.0e0
$
```

Notes on running this problem: Quadratic interpolation for density of fluid may cause problems at the interface. This one run with regular MPM implementation. Have to allow fluid particles to accumulate strain as opposed to regular fluid where strain rate is used and accumulation is not necessary. Hardwire mass of piston.

Box 12. First Input File for Piston-Container Problem

```

1.0d-3      !density
2010.0d0,0.0d0,10.d25,2.d25,1.0d25,1.d2,0.d0
3,3,3,8     !constitutive equation
20          !number of particles in the sequence
20,1        !total number of particles to read in; no. per cell
0.500000    39.500000  0.00   0.00   0.00   0.00   0.00
0.500000    38.500000  0.00   0.00   0.00   0.00   0.00
0.500000    37.500000  0.00   0.00   0.00   0.00   0.00
0.500000    36.500000  0.00   0.00   0.00   0.00   0.00
0.500000    35.500000  0.00   0.00   0.00   0.00   0.00
0.500000    34.500000  0.00   0.00   0.00   0.00   0.00
0.500000    33.500000  0.00   0.00   0.00   0.00   0.00
0.500000    32.500000  0.00   0.00   0.00   0.00   0.00
0.500000    31.500000  0.00   0.00   0.00   0.00   0.00
0.500000    30.500000  0.00   0.00   0.00   0.00   0.00
0.500000    29.500000  0.00   0.00   0.00   0.00   0.00
0.500000    28.500000  0.00   0.00   0.00   0.00   0.00
0.500000    27.500000  0.00   0.00   0.00   0.00   0.00
0.500000    26.500000  0.00   0.00   0.00   0.00   0.00
0.500000    25.500000  0.00   0.00   0.00   0.00   0.00
0.500000    24.500000  0.00   0.00   0.00   0.00   0.00
0.500000    23.500000  0.00   0.00   0.00   0.00   0.00
0.500000    22.500000  0.00   0.00   0.00   0.00   0.00
0.500000    21.500000  0.00   0.00   0.00   0.00   0.00
0.500000    20.500000  0.00   -1.d0  0.00   0.00   0.00

```

Box 13. Second Input File for Piston-Container Problem

A.4.11 Stability Test Problem (Section 8.5, page 141)

```
"ary"
"stability test problem " '
$in
nx=6, ny=2, iprint=0, igen=0,
xleft=0.00e0,xright=5.00e0,ybot=0.00e0,ytop=1.00e0,
tmax=0.03,tstep=0.0010,dtmax=0.001,
tol=1.e-4,miter=30,wl=0,wr=0,wb=1,wt=1,
gravc=000.0d0
$
$fluid
mu=0.00e0
$
$regrig
$
$reg
$
```

```
1.0d0          !density
1.0d6,0.00d0,10.d25,2.d25,1.0d25,1.d2,0.d0
3,3,3,0        !constitutive equation
5
5,1
0.0001  0.5  0.00  0.00  0.00  0.00  0.00
1.0001  0.5  0.00  0.00  0.00  0.00  0.00
2.0001  0.5  1.d-8  0.00  0.00  0.00  0.00
3.0001  0.5  0.00  0.00  0.00  0.00  0.00
4.0001  0.5  0.00  0.00  0.00  0.00  0.00
```

Box 14. Input Files for Stability Test Problem

A.4.12 Two-Bar Impact (Section A.1.5, page 160)

```
'"ary"'
'"2 bars impacting" '
$in
nx=21, ny=21, iprint=0, igen=1,
xleft=0.00e0,xright=5.000e0,ybot=-0.000e0,ytop=5.00e0,cntr=1.0,
tmax=0.501,tstep=0.0007e0,dtmax=0.0007,tplt=0.5,
tol=1.e-4,miter=30,large=.false.,wl=1,wr=1,wb=1,wt=1,
iout(13)=1,tiout(13)=0.1,
iout(30)=0,tiout(30)=1.d-1,
iout(31)=0,tiout(31)=1.d-1,
iout(32)=0,tiout(32)=1.d-1,
iout(33)=1,tiout(33)=0.005,
iout(34)=1,tiout(34)=0.0025,
iout(35)=0,tiout(35)=0.0007,
iout(36)=0,tiout(36)=0.01,
iout(37)=0,tiout(37)=2.5e-1,
iout(40)=0,tiout(40)=0.1,
methd=0,gravc=0.0
$
$fluid
$
$regrig
$
$reg
nreg=2,
nvi=4,4,
xvi(1,1)=1.0, 2.25,2.25,1.0,
yvi(1,1)=0.25,0.25,4.75,4.75,
xvi(1,2)=2.75,4.0, 4.0, 2.75,
yvi(1,2)=0.25,0.25,4.75,4.75,
xcenter=0.0,0.0, ycenter=0.0,0.0,
rix(1)=0.0d0, rex(1)=1.5e3,
riy(1)=0.0d0, rey(1)=1.5e3,
rix(2)=0.0d0, rex(2)=1.5e3,
riy(2)=0.0d0, rey(2)=1.5e3,
nprg=4,4,
densir(1)=1.0,1.0,
ipath(1,1)=3,3,3,2,
ipath(1,2)=3,3,3,2,
matr(1,1)=1.e5,0.30e0,10.e25,2.e25,1.0e25,1.e2,0.e0,
matr(1,2)=1.e5,0.30e0,10.e25,2.e25,1.0e25,1.e2,0.e0,
vrax(1)= 2.0e0,
vrax(2)= -0.0e0
$
```

Box 15. Input File 1 for Ball and Net Simulation

A.4.13 Ball and Net Simulation (Section A.1.6, page 164)

```
"ary"
"ball and net orig"
$in
nx=41, ny=41, iprint=0, igen=3,
xleft=0.00e0, xright=10.000e0, ybot=0.00e0, ytop=10.0e0, cntr=1.0,
tmax=4.02, tstep=0.001d0, dtmax=0.001d0, tplt=0.5,
tol=1.e-4, miter=30, large=.false., wl=0, wr=1, wb=0, wt=0,
iout(13)=1, tiout(13)=1.0,
iout(31)=0, tiout(31)=1.d-1,
iout(32)=0, tiout(32)=1.d-1,
iout(33)=0, tiout(33)=0.005,
iout(34)=1, tiout(34)=0.005,
iout(37)=0, tiout(37)=2.5e-1,
iout(40)=0, tiout(40)=0.1,
methd=0, gravc=0.0
$
$fluid
$
$regrig
$
$reg
nreg=1,
nvi=4,
xvi(1,1)=0.0e0, 10.0e0, 10.0e0, 0.0e0,
yvi(1,1)=0.0e0, 0.0e0, 10.0e0, 10.0e0,
xcenter=5.625, ycenter=5.0,
rix(1)=0.0d0, rex(1)=1.5,
riy(1)=0.0d0, rey(1)=1.5,
nprg=4,
densir(1)=1.e0,
ipath(1,1)=3,3,3,2,
matr(1,1)=1.e4, 0.30e0, 10.e25, 2.e25, 1.0e25, 1.e2, 0.e0,
vrx(1)= 1.0e0
$
```

Box 16. Input File 1 for Ball and Net Simulation

A.4.14 Maple Calculations

Input to Maple

```

Nj1:=N-dt/dx*( (vpm1+vpp1)*(N-N^2)+vp*(2.*N^2-2.*N+1.) );
Mj1:=1.-dt/dx*( (vpp1-vpm2)*(N-N^2)+(vp-vpm1)*(3.*N^2-3.*N+1.) );
d:=Nj1/Mj1;
Mjp1:=1.-dt/dx*( (vpp2-vpm1)*(N-N^2)+(vpp1-vp)*(3.*N^2-3.*N+1.) );
e:=(1.-Nj1)/Mjp1;
f:=vpm2*(-d)*(1.-N)+
vpm1*( d*(2.-3.*N)-e*(1.-N) )+
vp*( d*(3.*N-1.)+e*(2.-3.*N) ) +
vpp1*( -d*N + e*(3.*N-1.) )+
vpp2*(-e)*N;
vpm2:=0.0;
vpm1:=0.0;
vpp1:=0.0;
vpp2:=0.0;
simplify(f);
fH:=subs(N=0.5,f);
simplify(fH);
f0:=subs(N=0.0,f);
simplify(f0);
f1:=subs(N=1.0,f);
simplify(f1);

```

Output from Maple

```

|\^/|      MAPLE V
_|_|_|_|_|_ Copyright (c) 1981-1990 by the University of Waterloo.
\ MAPLE /   All rights reserved.  MAPLE is a registered trademark of
<____>     Waterloo Maple Software.
|           Type ? for help.
> read maple;

                2          2
      dt ((vpm1 + vpp1) (N - N ) + vp (2. N  - 2. N + 1.))
Nj1 := N - -----
                dx

                2          2
      dt ((vpp1 - vpm2) (N - N ) + (vp - vpm1) (3. N  - 3. N + 1.))
Mj1 := 1. - -----
                dx

                2          2
      dt ((vpm1 + vpp1) (N - N ) + vp (2. N  - 2. N + 1.))
N - -----
                dx

d := -----
                2          2
      dt ((vpp1 - vpm2) (N - N ) + (vp - vpm1) (3. N  - 3. N + 1.))
1. - -----

```

```

dx

Mjpl := 1. - 
$$\frac{dt \left( (vpp2 - vpm1) (N - N^2) + (vpp1 - vp) (3. N^2 - 3. N + 1.) \right)}{dx}$$


1. - N + 
$$\frac{dt \left( (vpm1 + vpp1) (N - N^2) + vp (2. N^2 - 2. N + 1.) \right)}{dx}$$


e := 
$$\frac{dt \left( (vpp2 - vpm1) (N - N^2) + (vpp1 - vp) (3. N^2 - 3. N + 1.) \right)}{1. - \frac{dt \left( (vpm1 + vpp1) (N - N^2) + vp (2. N^2 - 2. N + 1.) \right)}{dx}}$$


f := - 
$$\frac{vpm2 (N - \%3) (1. - N)}{\%4}$$


+ vpm1 
$$\left| \frac{/(N - \%3) (2. - 3. N)}{\%4} - \frac{(1. - N + \%3) (1. - N)}{\%2} \right|$$


+ vp 
$$\left| \frac{/(N - \%3) (3. N - 1.)}{\%4} + \frac{(1. - N + \%3) (2. - 3. N)}{\%2} \right|$$


+ vpp1 
$$\left| - \frac{(N - \%3) N}{\%4} + \frac{(1. - N + \%3) (3. N - 1.)}{\%2} \right| - \frac{vpp2 (1. - N + \%3) N}{\%2}$$


%1 := 
$$3. N^2 - 3. N + 1.$$


%2 := 
$$1. - \frac{dt \left( (vpp2 - vpm1) (N - N^2) + (vpp1 - vp) \%1 \right)}{dx}$$


%3 := 
$$\frac{dt \left( (vpm1 + vpp1) (N - N^2) + vp (2. N^2 - 2. N + 1.) \right)}{dx}$$


%4 := 
$$1. - \frac{dt \left( (vpp1 - vpm2) (N - N^2) + (vp - vpm1) \%1 \right)}{dx}$$


vpm2 := 0

vpm1 := 0

vpp1 := 0

vpp2 := 0

```

$$\begin{aligned} & \text{vp} (6. \text{dt}^2 \text{vp}^2 \text{N}^4 - 6. \text{N}^2 \text{dx}^2 + 6. \text{N}^2 \text{dx}^2 + \text{dt}^2 \text{vp}^2 - 2. \text{dx}^2 + 2. \text{N} \text{dx} \text{dt} \text{vp} \\ & - 12. \text{dt}^2 \text{vp}^2 \text{N}^3 + 11. \text{dt}^2 \text{vp}^2 \text{N}^2 - 5. \text{dt}^2 \text{vp}^2 \text{N} - 1. \text{dt} \text{vp} \text{dx}) / (\\ & / \end{aligned}$$

$$(-1. \text{dx} + 3. \text{dt} \text{vp} \text{N}^2 - 3. \text{dt} \text{vp} \text{N} + \text{dt} \text{vp})$$

$$(\text{dx} + 3. \text{dt} \text{vp} \text{N}^2 - 3. \text{dt} \text{vp} \text{N} + \text{dt} \text{vp}))$$

$$\text{fH} := \text{vp} \left[\begin{array}{c} / \\ \left| \begin{array}{cc} .5 - .50 \frac{\text{dt} \text{vp}}{\text{dx}} & .5 + .50 \frac{\text{dt} \text{vp}}{\text{dx}} \end{array} \right| \\ .5 \frac{\text{dt} \text{vp}}{\text{dx}} + .5 \frac{\text{dt} \text{vp}}{\text{dx}} \\ \left| \begin{array}{cc} 1. - .25 \frac{\text{dt} \text{vp}}{\text{dx}} & 1. + .25 \frac{\text{dt} \text{vp}}{\text{dx}} \end{array} \right| \\ \backslash \end{array} \right]$$

$$2. \frac{\text{vp} (-4. \text{dx}^2 + \text{dt}^2 \text{vp})}{(-4. \text{dx} + \text{dt} \text{vp}) (4. \text{dx} + \text{dt} \text{vp})}$$

$$\text{f0} := \text{vp} \left[\begin{array}{c} / \\ \left| \begin{array}{c} 1. \frac{\text{dt} \text{vp}}{\text{dx}} + 2. \\ \left| \begin{array}{c} / \\ \text{dx} \left| 1. - 1. \frac{\text{dt} \text{vp}}{\text{dx}} \right| \\ \backslash \end{array} \right| \end{array} \right| \end{array} \right]$$

$$\frac{\text{vp} (\text{dt} \text{vp} - 2. \text{dx})}{-1. \text{dx} + \text{dt} \text{vp}}$$

$$\text{f1} := \text{vp} \left[\begin{array}{c} / \\ \left| \begin{array}{cc} 1.0 - 1. \frac{\text{dt} \text{vp}}{\text{dx}} & \end{array} \right| \\ 2.0 \frac{\text{dt} \text{vp}}{\text{dx}} - 1.0 \frac{\text{dt} \text{vp}}{\text{dx}} \\ \left| \begin{array}{cc} 1. - 1. \frac{\text{dt} \text{vp}}{\text{dx}} & \text{dx} \left| 1. + 1. \frac{\text{dt} \text{vp}}{\text{dx}} \right| \end{array} \right| \\ \backslash \end{array} \right]$$

$$\frac{\text{vp} (2. \text{dx} + \text{dt} \text{vp})}{\text{dx} + \text{dt} \text{vp}}$$

Distribution

1	MS0457	H.W. Schmitt	2000
1	MS0429	R.D. Andreas	2100
5	MS0483	A.R. York	2165
1	MS0453	N.R. Hansen	2104
1	MS0437	R.K. Thomas	9118
1	MS0437	S. W. Attaway	9118
1	MS0437	F.J. Mello	9118
1	MS0437	M.R. Tabbara	9118
1	MS0841	P.J. Hommert	9100
1	MS0443	H.S. Morgan	9117
1	MS0443	M.W. Heinsteins	9117
1	MS0820	P. Yarrington	9232
1	MS1175	D.S. Oscar	9364
1	MS9018	Central Technical Files	8940-2
5	MS0899	Technical Library	4916
2	MS0619	Review & Approval Desk	12690
		For DOE/OSTI	

Walter Zhou
 Dept. of Mechanical Engineering
 University of New Mexico
 Albuquerque, NM 87131

Dr. Lee Davison
 7900 Harwood Ave NE
 Albuquerque, NM 87110

Dr. M. Ingber
 Dept. of Mechanical Engineering
 University of New Mexico
 Albuquerque, NM 87131

Dr. H.L. Schreyer (2)
 Dept. of Mechanical Engineering
 University of New Mexico
 Albuquerque, NM 87131

Dr. Deborah Sulsky (2)
 Dept. of Mathematics and Statistics
 University of New Mexico
 Albuquerque, NM 87131

Distribution

Dr. Jeremiah Brackbill
T-3, B216
Los Alamos National Laboratory
PO Box 1663
Los Alamos, New Mexico 87545

Renida R. Carter
ESA-EA P946
Los Alamos National Laboratory
PO Box 1663
Los Alamos, New Mexico 87545

Donna Hale
(208)526-1744
MS3765
INEEL
PO Box 1625
Idaho Falls, ID 83415

Dr. Qiuhai "Ken" Zuo
Theoretical Division/Fluid Dynamics Group (T-3)
Mail Stop B216
Los Alamos National Laboratory
Los Alamos, New Mexico 87545